Visual Thesaurus for Color Image Retrieval using SOM

YIP King-Fung

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of

Master of Philosophy

in

System Engineering and Engineering Management

Janurary 2003

Abstract

of Visual Thesaurus in Color Image Retrieval using SOM

Submitted by YIP King-Fung

for the degree of Master of Philosophy

in System Engineering and Engineering Management

at The Chinese University of Hong Kong in August 2002

The technique of searching in content-based image retrieval has been actively studied in recent years. However, this technique has multiple drawbacks. For example, one is the problem of selecting the initial example image and another disadvantage is the possibility of being confined to a specific set of search results. For the present study, a novel browsing technique has been proposed using Kohonen's Self-Organizing Map (SOM) to enhance the effectiveness of retrieval for a general color image database. In this project, both the chromatic and textural feature of images are analyzed in order to represent the content of images. In the current research, initially a prototype system was established which generates SOMs according to the image features. Secondly, results were compared with different image features and labeling methods. Finally, human evaluation was used to compare the performance between this browsing technique and the traditional query-by-example (QBE) technique. Empirical results show that our SOM approach outperforms the traditional QBE approach in terms of (1) higher efficiency; (2) higher successful rate; and (3) encouraging more queries. The survey also indicates that users are more satisfied in using SOM rather than QBE.

In addition, an algorithm has been established to quantize non-cubical color space like CIELUV and CIELAB for color histogram indexing. The performance was

compared by indexing effectiveness and human evaluation.

Finally, an approach has been proposed that will employ relevance feedback in visual thesaurus. A prototype of the user interface has also been developed that will be used to test its applicability.

論文摘要

彩色影像檢索的視覺索引--使用自我映射組織圖

由葉勁峰提交

香港中文大學系統工程及工程管理哲學碩士學位

二零零二年八月

近年，基於內容的影像檢索中的搜尋技巧已被廣泛地研究。可是，這種搜尋技巧卻有多個缺點，例如選擇最初的搜尋例子的問題、被困在一組搜尋結果的可能性等等。這研究建議一種新的瀏覽技巧，利用 Kohonen 的自我映射組織圖(Self-Organizing Map, SOM)去有效地檢索一般彩色影像資料庫。影像的顏色及紋理特徵經分析後去代表影像的內容。在這研究裡，建立了一個原型系統利用那些影像的特徵去產生自我映射組織圖，並且比較不同的影像特徵及標籤方法的結果。之後，利用使用者評估的方法的比較 SOM 技巧及傳統的以例子搜尋(QBE)技巧。結果影示 SOM 技巧比傳統的 QBE 方法有效，包括 (1) 較高的效率、(2)較高的成功率及(3)鼓勵更多的提出。

另外，這研究建立了一個演算法去量化非立方形的顏色空間，例如 CIELUV 及 CIELAB 顏色空間，作爲顏色分佈圖之用，並透過索引有效性及使用者評估的方法的比較它的表現。

最後，這研究建議一個方法把關聯回饋加入視覺索引中，並透過發展一個使用者介面的原形去測試它的可應用性。

Table of Contents

List of Abbreviations

CBIR	Content-Based Image Retrieval
CBIQ	Content-Based Image Query
CIE	Commission Internationale de L'Éclairage
CMY	Cyan-Magenta-Yellow (color space)
GLA	General Loyld Algorithm
LHS	Luminance-Hue-Saturation (color space)
HSV	Hue-Saturation-Value (color space)
HLS	Hue-Luminance-Saturation (color space)
MARS	Multimedia Analysis and Retrieval Systems (CBIR system)
MR-SAR	Multiresolution Simultaneous Autoregressive Model
PWT	Pyramid-structured Wavelet Transform
QBE	Query By Example
QBIC	Query By Image Content (CBIR system)
RGB	Red-Green-Blue (color space)
SOM	Self-Organizing Map
VQ	Vector Quantization

Acknowledgements

The research presented in the thesis has been carried out at the Department of System Engineering and Engineering Management (SEEM) at The Chinese University of Hong Kong.

First of all, I would like to express my deepest gratitude to my supervisor, Prof. Christopher C. Yang, for providing excellent ideas, guidance, expert opinions and patience throughout the two-year research.

I would like to thank Mandy C. Chan, a former student of my supervisor at The University of Hong Kong, for demonstrating and explaining the former CBIR project in detail. Also, I would like to thank all staff in SEEM and especially my colleagues including Jaffe Li, Philips Wong, Eddy Lau and Shirley Kwok, for giving their hands providing valuable assistance in many various ways.

Finally, I would like to thank my parents and my wife Jovey Chan for all their whole-hearted support during my studies.

1. Introduction

1.1. Background

Recently, many systems for content-based image retrieval (CBIR) have been developed. Examples of these are IBM's Query by Image Content (QBIC) [Niblack93, Flickner95, Hafner95, Niblack98], and the University of California, Berkeley's Digital Library project [Berkeley]. Most of these methods use the technique of query-by-example, a searching approach that is used to find images that are similar to the given example image. Similarity between images is quantified in terms of some global or local features, such as color, texture, shape, pattern, or specific spatial area of those previously mentioned features. The feature selection and matching techniques are vital to searching, thus considerable research has been carried out that is based on these techniques. However, there are some serious limitations associated with exclusive use of the searching approach. For example, query-by-example techniques often generate results of a relatively small amount of images, which may not be of interest to the user. Consequently, the user may not be able to continue further with querying. By contrast, the browsing environment offers an alternative approach with which to tackle the problem. However it has attracted little attention.

According to Craver et al. [Carver98], browsing is a technique, or a process, that allows the users to view information rapidly. Subsequently, the user can decide whether or not the content is relevant to the application. In this way the user can obtain an overview or summary of contents quickly and then focus on particular sections of interest. Craver et al. modeled a general process of image browsing that consisted of several stages. The first step is to extract the relationship among the

images. The second step is to find representative instances of images during browsing. And the final step is to visualize the images with intuitive presentation of the relationship among images.

Recent research in the area of images browsing has used different types of data structure to organize images in a meaningful way in order to present the relationship among images. Craver et al. has described a technique based on multiple space-filling curves. Chen et al. [Chen99a] initiated a new approach called active browsing, which employed their previous work on similarity pyramids. These techniques provide attractive features. For example the process of inserting new images is computationally fast and in addition, complete re-indexing is not necessary. However, these techniques have their limitations. A single space-filling curve would result in a set of similar images that were sparsely located. Despite the fact that use of dual or multiple space-filling curves will alleviate the problem slightly, it will increase the complexity of user-interface accordingly. Active browsing uses a quad-tree structure for storing four similar images under a node. However, the arrangement of images is fragmented because the images in the boundaries of sibling nodes are dissimilar.

Another technique for image browsing utilizes a type of unsupervised artificial network. This has been referred to as Kohonen's self- organizing map (SOM) [Kohonen84, Kohonen95]. The significant feature of SOM is that it can reduce high-dimensional input signal space to low-dimensional space while the map preserves the relationship among the input signals. For visualizing purpose, most applications select two-dimensional grid form SOM.

The SOM algorithm has been applied to the field of content-based image retrieval. Zhang et al. [Zhang95] addressed the importance of developing effective indexing scheme for image database. The study focus was an indexing scheme with application to the SOM approach. In their first experiment, three texture features, namely, SAR, coarseness and gray – were used for indexing monochromatic textural images. In their next experiment, a color histogram was used for indexing general color images. For the purpose of the study, a set of hierarchical self-organizing maps (HSOMs) was developed in order to construct an index tree, which provides a space for searching. The performance was evaluated by retrieval rate.

Another research project was carried out by Han et al. [Han95]. That study also used SOM for image retrieval. However, the database was restricted to images of objects, from which shape features such as roundness, rectangularity, elipticity, eccentricity and bending energy were extracted. Only the luminance information was considered.

In the present thesis, an approach will be presented to provide an image-browsing interface to enable personnel to retrieve general color image database using SOM [Yang01]. Experiments have been conducted to compare the performance between this approach with the traditional Query-by-example approach.

1.2. Motivation

The purpose of this study is to demonstrate that visual thesaurus is an important and widely applicable technique in image retrieval. In this thesis, specific architecture, algorithms and prototypes have been created for use with the image thesaurus in CBIR. In addition, there has been a thorough investigation of the performance of the novel approach in comparison with the traditional approach. During the present

research, there has also been an investigation of a new technique for color histogram, which can be applied to general CBIR systems.

1.3. Thesis Organization

Chapter 2 reviews previous research in the area of image retrieval by discussing various approaches and systems. The proposed visual thesaurus approach is presented in Chapter 3. In Chapter 4, there will be a discussion of the experiment investigating the performance between the present approach and that of the traditional approach. Chapter 5 presents the newly proposed quantizing color histogram, which can be used in general CBIR systems. Studies on embedding relevance feedback technique for the visual thesaurus approach are provided in Chapter 6. Finally, the conclusions and future directions are presented in Chapter 7.

2. A Survey of Content-based Image Retrieval

Content-based Image Retrieval (CBIR) is a technique that utilizes the visual content of the images in the process of retrieving images from an image database. The aim of this system is the use of differences that corresponded to human judgment of image similarity in the retrieval process. In CBIR, images are indexed by features directly derived from visual content of the images. These features are usually low-level information of the images, such as color, texture, shape, and combinations of above those features.

In this chapter there will be detailed discussion on various topics in the area of image retrieval. First, two main retrieval approaches, text-based and content-based image retrieval, are presented. Second, a review on various well-known CBIR systems will be given.

2.1. Text-based Image Retrieval

For decades, the traditional approach to image retrieval was based on manual input of keywords. The keywords described the contents of image and other relevant information such as when and where the image was taken. The user formulated textual queries which were then used to search against the keywords. The advantage of this approach was that it enabled widely approved text information retrieval systems to be used for visual retrieval systems.

However, there are some serious problems that have been associated with this textual annotation approach. First, manual input of keywords is required. However it is not the time and cost involved in this process that is a problem. There are other problems

caused by manual annotation. For example, it is problematic to fully describe an image. It is the responsibility of the annotator personnel to provide an accurate description of the characteristics of every object in addition to their spatial relationships, and relationships among other objects in the image. It is usually impossible to describe an image fully when the image consists of many objects or details. Queries to these images may also generate a large number of irrelevant results. This approach also becomes impractical as the image database grows in size.

The second problem with this approach is that descriptions of images are very often subjective. Different people have different interpretation of an image, such as its important objects, or relationships. Annotators will also face serious difficulties in maintaining the consistency of annotation among images in large databases. To retrieve a particular image that he/she wants, a user must know the exact terms that the annotator has used, which is basically impossible if the user has not been trained to understand the annotation generation process and the underlying rules that apply.

Although the above problems that have arisen in text-based image retrieval approach have been encountered, text information is still important since the present system of automatic image content extraction is far from perfect and insufficient.

One recent example of using text-based information in image retrieval is Google [Google]. The Google system has been developed as an image search engine by adaptation of their technology for use in full-text searching. The images in their database are indexed by the proximity of the text to the image. This approach is completely automatic; nevertheless, the text description that is extracted may not describe the corresponding image accurately. There is a lot of extraneous information

that would affect the effectiveness of retrieval. Other search engines have adopted a similar approach [Swain99], for example, WebSeer [Frankel96], AltaVista Photo Finder [AltaVista].

2.2. Content-Based Image Retrieval

Content-Based Image Retrieval has been a subject for research for a long period of time. However, due to the increased demand for practical applications and the serious limitations of the manual annotation paradigm, this particular field of research field has become very active in recent years.

In the application of Content-Based Image Retrieval, the images are indexed by features that are derived directly from the images. The features are always consistent with the image and they are extracted and analyzed automatically by means of the computer, instead of manual annotation.

Due to the difficulty of automatic object recognition, information extracted from images in CBIR is rather low-level, such as colors, textures, shapes, and the combinations of the above. These features should correlate with human judgment of similarity as much as possible.

2.2.1. Content-Based Image Retrieval Systems

In this section, a number of representative generic CBIR systems will be reviewed. These systems have been implemented in different environments, some of which are Web-based while some are GUI-based application.

QBIC

Developed at the IBM Almaden Research Centre, QBIC is one of the best-known systems for CBIR [Flickner95, Hafner95, Niblack93, Niblack98]. QBIC was the first commercial CBIR application and consequently it played a vital role in the evolution of the whole image retrieval research field.

The QBIC system supports low-level image features of average color, color histogram, color layout, texture and shape. Additionally, users can provide pictures or draw sketches as example images in query. The visual queries can also be combined with textual keyword predicates.

Photobook

Yet another example is MIT Media Lab's Photobook [Pentland94] which is a set of interactive tools for searching and querying images. It is divided into three specialized systems, namely Appearance Photobook (face images), Texture Photobook, and Shape Photobook, which can also be used in combination. The features are compared by using one of the matching algorithms. These include Euclidean, Mahalanobis, divergence, vector space angle, histogram, Fourier peak, and wavelet tree distances, as well as any linear combination of those previously discussed.

NETRA

This system, NETRA is a prototype image retrieval system that has been developed at the University of California, Santa Barbara (UCSB) [Manjunath97]. NETRA supports features of color, texture, shape, and spatial information of segmented image regions to query similar regions in the database.

A new automated image segmentation algorithm has been developed for region-based search. In this particular system, images are segmented to homogenous regions when they are inserted to the database, and the features are generated for every region of the images. Using the region as the basic unit, users can submit queries based on features that combine regions of multiple images. For example, a user may compose queries such as "retrieve all images that contain regions having color of region of image A, texture of a region of image B, shape of a region of image C".

2.2.2. Query Methods

Image queries in CBIR systems are usually performed by using an example image or series of images. The task of the system is to determine which images are the most similar to the given images. This approach is generally called Query by Pictorial Example (QBPE) or simply Query by Example (QBE). The retrieval interaction begins with an initial selection of reference images. The initial selection can be randomly selected images or some representative images selected by any means. Subsequently, the user can choose one of the images and the system will retrieve those images that are most similar to the reference. One limitation of QBE is that the success of query depends heavily on the initial set of images. In large databases, finding a set of initial images that contains at least one relevant image can be problematic. La Cascia et al. [LaCascia98] have described this situation as a page zero problem.

To overcome this dilemma, some CBIR systems allow users to provide separate images outside the database. Some systems also provide a canvas to enable users to

sketch an image for query. These approaches require the system to index the separate images online in order to evaluate the similarity between the query image and the images in database.

Additionally, some systems support natural language queries [Haradar97], which may be especially useful for inexperienced users with no knowledge of the underlying mechanism of the system. Even when using a database that does not have text annotations for the images, the users can formulate traditional database queries, for example, "blue flower", "sunrise and sunsets", etc. However, it is extremely difficult to relate and evaluate the similarity of the semantics between the natural language queries and images.

Evaluating the similarity between different images is the fundamental task in CBIR. To be effective, a useful similarity measure should produce large value for similar images, small value for dissimilar images. As the objective of CBIR is for finding images relevant to a specific user, the similarity should correspond to the concept of similarity of user. Thus, the performance of the CBIR systems is dependent on the measurement.

Most traditional textual information retrieval systems are based on the Boolean model for queries. Use of this model suggests that if a document is relevant then there is a keyword in it that matches the query. However, this model is not applicable to CBIR systems because it is difficult to find appropriate binary matching criteria to pick up only the desired images. Therefore, the relevant images are sorted according to the similarity measurement selected as a result of the query. One definition of Content-Based Image Query (CBIQ) from Smith [Smith97] is as follows:

Content-Based Image Query: *Given an image database D of N images and a feature dissimilarity function d(I, J), find the* N_{cutoff} *images* $J \in D$ *with the lowest dissimilarity d(I, J) to the query image I.*

Another definition refers to returning images *J* which have lower dissimilarity to *I* than a certain threshold.

Use of one single feature may not correspond to the user requirement in querying. An improvement on this system can be achieved by combining the dissimilarity powers of multiple features in order to increase the evaluation effectiveness.

2.2.3. Image Features

One of the main foci in CBIR is the means for extraction of the features of the images and evaluation of the similarity measurement between the features. Image features refer to the characteristics which describe the contents of an image. In this thesis, image features are confined to visual features that are derived from image directly.

There has been extensive studied of various sorts of features, because of the importance of visual feature as the foundation of all kinds of applications of CBIR. Most works in this area have concentrated on finding the best features and indexing methods to represent the similarities among images.

The simplest form of visual features is directly based on pixel values of the image [15]. However, this type of visual feature is very sensitive to noise, brightness, hue

and saturation changes, and are not invariant to spatial transformations such as translation and rotations. As a result, CBIR systems that are based on pixel values do not generally have satisfactory results. Much of the research in this area has placed the emphasis on computing useful characteristics from images using image processing and computer vision techniques.

Usually, general-purpose features in CBIR have included color, texture, shape and structure. Other features are specific to the application domains and require some special knowledge and consequently put constraints on the database. For example, facial CBIR systems require techniques widely studied in image processing for face recognition. In this thesis, the aim is to concentrate on general-purpose features.

The representation of the content of an image I is usually compiled into a d-dimensional feature vector $\mathbf{f}^I$:

$$\mathbf{f}^I = \begin{pmatrix} f_1^I & f_2^I & f_3^I & \cdots & f_d^I \end{pmatrix}^T \tag{2-1}$$

The dimensionality d of the feature vector directly affects the performance of image query. A typical value of d in the context CBIR is of the order 100 [16]. In the simplest form of query processing without indexing, $O((Nd)^2)$ computations are required to compare each element of all vector pairs, where N is number of the images.

The measurement of similarity between two feature vectors is commonly based on distance. Some distance metric can be used for $\mathbf{f}^I$ and $\mathbf{f}^J$ to measure dissimilarity between two images I and J. A commonly used choice is Euclidean or L_2 norm:

$$D_{L_2}(I,J) = \sqrt{\sum_{i=1}^{d} (f_i^I - f_i^J)^2} \quad (2\text{-}2)$$

Other metrics, such as L_1 norm, are used in some applications. Weighted distance metrics may also be a viable choice. The efficiency of measurement shall be carefully considered for excessively large database or online applications.

Understandably, the choice of relevant and suitable features is the key issue when designing CBIR systems. A good feature should contain sufficient discriminating power to distinguish between similar and dissimilar images. Also, features should be invariant to spatial transformation such as translation, rotation, and minor changes related to the lighting environment where the image is captured.

Chromatic Features

Color is a simple and straightforward feature of general color images. Color is a psychophysical phenomenon for human vision. Human visual systems are more sensitive to levels of hue than levels of gray. The color characteristics in images are often an important element of the image content. Many common materials and backgrounds have distinct color properties, for example, grass is green, sea is blue, and human skin has a series of distinguishable colors.

Color Spaces: Color representation is based on the classical work of Thomas Young, who stated that any color could be reproduced by mixing three primary colors. Later research verified that the retina contains three types of color receptors with different absorption spectra. A color space is a part of a three dimensional coordinate system

where a color is represented as a vector. Selection of an appropriate color space is the starting point of using color feature in CBIR systems.

The most frequently used color space used in computer technology is RGB because it is the common color space for CRT (Cathode Ray Tube) which uses red, green and blue phosphor-coated screen to reproduce color images. The RGB color space has a very simple geometry. The red, green and blue components are orthogonal axes. The color space is a cube-shape.

RGB is an additive color space where the lights of three components are combined together to reproduce color. Another common color space is CMY, which has subtractive primary colors: cyan, magenta and yellow. It is used mostly in printers, where inks will subtract the reflective strength of components.

One problem of RGB and CMY is that the components are irrelevant to human visual perception which normally interpret color as hue, saturation and luminance. Therefore it is difficult for the user to select a color by changing the level of the axes. Also, RGB and CMY color spaces is not suitable in some applications because they are not perceptually uniform. The distance between two colors is not in proportion to the level of similarity between the colors.

LHS attempts to model the human visual system's perceptual response to luminance, hue and saturation. It is derived from the Maxwell triangle in the RGB space. HSV and HLS have an advantage of fast transformation from RGB color space. They are hexagon model and double hexagon model respectively.

CIELAB (CIE L*a*b*) and CIELUV (CIE L*u*v*) are two uniform color spaces developed by CIE (Commission Internationale de L'Éclairage). One of their limitations is that the transformation is computationally more expensive.

Average Color: Average color is the simplest representation of color features, which is calculated by averaging colors of all pixels in an image. It is generally considered too inaccurate for use in color image retrieval because it does not consider the distribution of colors.

Color histogram: Color histogram is the standard representation of color feature in CBIR system, initially investigated by Swain et al [Swain91] in 1991. This method uses histograms of intensity values to represent the color distribution. This kind of histogram captures the global chromatic information of an image. The advantage of this method is that the histogram is invariant under translation and rotation about the view axis. Despite changes in view, change in scale, and occlusion, the histogram only changes slightly.

The formal description of color histogram and the innovative approach used in this study to quantize color histogram is provided in chapter 5.

Texture Features

As the literature has indicated, texture analysis research has a long history. According to Manjunath et al. [Manjunath96], texture analysis algorithms have used various techniques, including random field model and multiresolution filtering techniques such as wavelet transform. For example, the research of Manjunath et al has compared the experimental results of a number of texture features. The texture

features included the conventional pyramid-structured wavelet transform (PWT) features, tree-structured wavelet transform (TWT) features, the multiresolution simultaneous autoregressive model (MR-SAR) features and the Gabor wavelet features. The experimental results demonstrated the superiority of the Gabor features, which give the best performance.

In another recent research carried out by Chen et al. [Chen99b], they compared four filtering techniques including Fourier transform, spatial filter, Gabor filter and wavelet transform for texture discrimination. The experimental results also showed that Gabor filter generally gives the best performance. However, Chen et al. stated that the execution time of Gabor filter is the longest out of the four features.

Since texture analysis is performed once only in this kind of image retrieval application, the disadvantage of longer execution time of generating Gabor features is not critical. Therefore, Gabor filter has been adopted for textural analysis in this research due to its superior performance. Section 3.1.1 will review the Manjunath et al.'s method of using Gabor filter for texture feature extraction.

2.2.4. Summary

In this chapter, the major problems, approaches, and systems in CBIR research field have been reviewed. Image feature is the fundamental problem in CBIR which has been studied extensively during the past 2 decades. However, most systems use image features for indexing and query purposes. In this thesis, the focus will be on use of image features in browsing techniques and related human-computer interaction issues.

3. Visual Thesaurus using SOM

3.1. Algorithm

The proposed algorithm can be divided into two main sections: image representation and self-organizing map generation. The former section transforms the image database into representation, which can be used for evaluating similarity among the images. The latter section generates SOM using the above information.

3.1.1. Image Representation

There are a number of color coordinate systems for images. Traditionally color images are represented in RGB format when processed in computer systems. However, this color coordinate system does not match with human perception of color. Color perception researchers generally believe that human recognize color by hue and saturation. Since this research is aimed at providing a user interface for human, LHS color coordinate system have been adopted as it is close to human color perception.

The transformation from RGB to LHS coordinate system is reviewed as follows.

$$L = 0.299R + 0.587G + 0.114B$$

$$H = \theta + \cos^{-1}\left(\frac{N}{\sqrt{6\left[(r-\frac{1}{3})^2 + (g-\frac{1}{3})^2 + (b-\frac{1}{3})^2 \right]}} \right)$$

$$S = 1 - \frac{3\min(R,G,B)}{R+G+B}$$

where

$$r = \frac{R}{R+G+B}$$

$$g = \frac{G}{R+G+B}$$

$$b = \frac{B}{R+G+B}$$

$$\theta = \begin{cases} 0^\circ \text{ if } b = \min(r,g,b) \\ 120^\circ \text{ if r} = \min(r,g,b) \\ 240^\circ \text{ if g} = \min(r,g,b) \end{cases}$$

$$N = \begin{cases} 2r - g - b \text{ if } b = \min(r,g,b) \\ 2g - r - b \text{ if r} = \min(r,g,b) \\ 2b - r - g \text{ if g} = \min(r,g,b) \end{cases}$$

The process of the proposed algorithm for image representation is shown as follows.

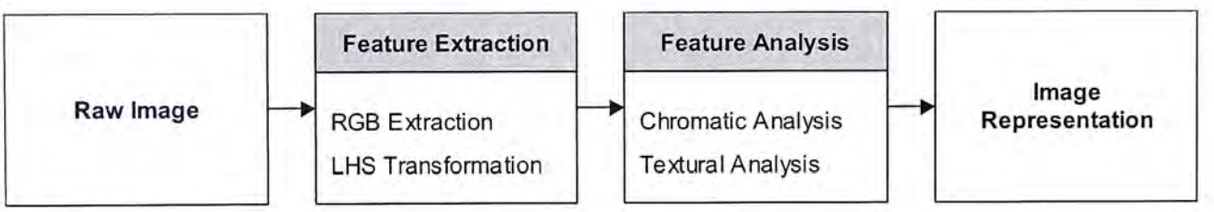


Figure 3-1. The process for image representation

Feature Extraction: In order to extract the content of images in the database, raw data of images are converted into some presentable and comparable features. There are various kinds of features of images, for example, color, texture, shape and pattern.

In this thesis, color and texture features are selected because the type of database is general color images [Yang99], while shape and pattern features are not suitable. For the color feature, hue and saturation component in LHS color coordinate system are used. Although all three components are necessary to represent a color, luminance of LHS space is omitted in the color feature. It is because images captured in the same

scene at different angles will affect its hue and saturation distribution only slightly but luminance distribution will be easily affected by lighting of the environment. Besides, the texture feature is analyzed by the luminance component only. Therefore, the process is denoted as:

For each image,

1. Extract RGB values of each pixel in the image
2. Convert the RGB values into LHS color coordinate system
3. Perform textural analysis with the L value
4. Perform chromatic analysis with the H and S values
5. Combine both analysis result to represent the image feature

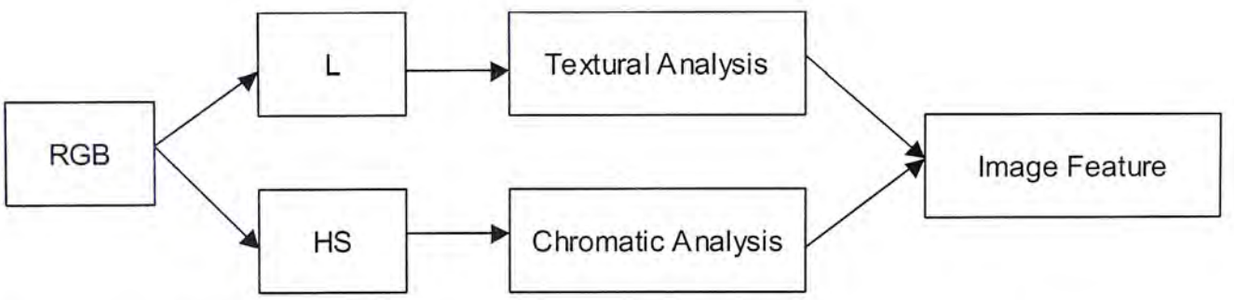


Figure 3-2. The information flow of the process for image representation

Chromatic Analysis: As mentioned before, H and S values for chromatic analysis are used. The algorithm is to build a two-dimensional histogram with one axis for hue and another for saturation. The definition of HS histogram is:

$$Histogram_{h,s}[h,s] = N \cdot \Pr(H = h, S = s) \tag{3-1}$$

where H and S is the hue and saturation channels, N is the number of pixels in the image.

The hue component varies from 0 to 360 degrees and the saturation component varies from 0 to 1. In order to build the histogram, the hue and saturation values are quantizied into several levels. 10 levels for each component are selected, therefore a 10 by 10 two-dimensional histogram $h[10,10]$ for each image is built in the present system.

A Pixel with $0° \le h < 36°$ and $0 \le s < 0.1$ is sorted in the first bin $h[1,1]$ and so on.

Textural Analysis: As section 2.2.3 reviewed different texture features, Gabor filter [Manjunath96] has been adopted in the present thesis. The computation of Gabor filter as texture feature is given as follows.

A two-dimensional Gabor function can be written as:

$$g(x,y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right)\exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)+2\pi jWx\right] \tag{3-2}$$

Then a self-similar filter dictionary can be obtained as a mother Gabor Wavelet *G(x, y)* by appropriate dilations and rotations of Eq. (3-2) as:

$$G_{mn} = a^{S-m}G(\theta_x,\theta_y) \tag{3-3}$$

where h = height of image, w = width of image,
hside = (h -1) / 2, wside = (w - 1) / 2
$\theta_x = (x - hside)\cos(n\pi/K) + (y - wside)\sin(n\pi/K)$
$\theta_y = -(x - hside)\sin(n\pi/K) + (y - wside)\cos(n\pi/K)$
$a > 1$, m,n are integers

Given an image with luminance $I(x,y)$, a Gabor decomposition can be obtained by multiplying the luminance by the magnitude of the Gabor Wavelet:

$$\left|W_{mn}(x,y)\right| = I(x,y)\sqrt{G_{mn}i^2 + G_{mn}r^2} \tag{3-4}$$

The mean and standard deviation of the magnitude of the transform coefficient are used to represent the texture feature for classification and retrieval purpose:

$$\mu_{mn} = \frac{\iint\left|W_{mn}(x,y)dxdy\right|}{h \cdot w} \tag{3-5}$$

$$\sigma_{mn} = \sqrt{\iint\left(\left|W_{mn}(x,y)\right| - \mu_{mn}(x,y)\right)^2 dxdy} \tag{3-6}$$

The Gabor feature vector is constructed by using μ_{mn} and σ_{mn} as feature components:

$$\bar{f} = \left[\mu_{00} \quad \sigma_{00} \quad \mu_{01} \quad \sigma_{01} \quad \cdots \quad \mu_{(S-1)(K-1)} \quad \sigma_{(S-1)(K-1)}\right] \tag{3-7}$$

where S is number of scales and K is number of orientation. In the following experiment, S=3 and K=4 are used.

3.1.2. Self-Organizing Map

After generating image representation, it is required to visualize the representation of all images in database by categorization. Categorization is a process which involves

grouping items of similar nature. There are several clustering algorithms available, such as K-means algorithm, single-link clustering and complete-link clustering (based on minimum-spanning tree). Kohonen [Kohonen84, Kohonen95] develops a connectionist approach called Self-Organizing Map (SOM) in 1981. SOM is supposed to reduce the high dimensional input space into lower dimensional space, usually two-dimensional. The resultant feature mapping is a non-linear projection from the input space. The topologically close nodes in the map are sensitive to the inputs that are similar.

Structure: According to Kohonen, the structure of SOM is typically an array of nodes (neurons) arranged in two-dimensional space. The nodes can be arranged in various ways, for example, rectangular, hexagonal, or even irregular. Rectangular arrangement is used in this research, as it is most suitable for visualization of a set of images. The structure of SOM used in this research is illustrated as follows.

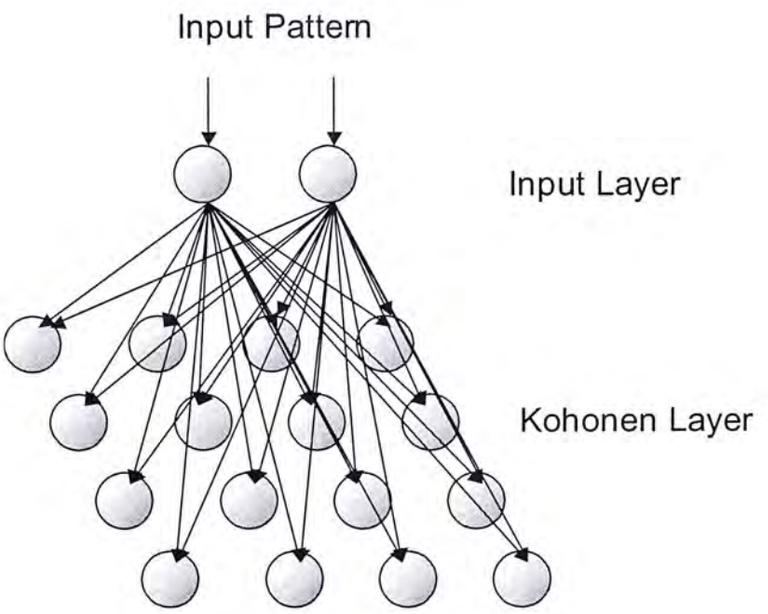


Figure 3-3. The structure of 2D SOM

The input patterns are the feature vectors of images in the database. Each node in

input layer (component of feature vector) is fully connected to the Kohonen Layer.

Training: The notations and training algorithm of SOM used in this research is as follows:

I Set of all images in database

x_i Feature vector of image $i \in I$

M Set of all nodes in Kohonen Layer

$w_j(t)$ Weight vector of node $j \in M$ *at time t*

R_j Set of images that are mapped to node $j \in M$

v_j Label image of node $j \in M$

1. Randomize $w_j(0), \forall j \in M$
2. For each iteration,
   - 2.1 Shuffle the presentation order of images
   - 2.2 For each image in the ordering list,
     - 2.2.1 Find a winning node c
     - 2.2.2 Update the winning node c and its neighbors
   - 2.3 Continue training with decreased learning rate and reduced area of neighborhood function
3. For each $i \in I$, put i into R_j

Firstly, random values are assigned to the weights of all nodes.. If there are identical weights of node, there may be a case that there are two winning nodes. To prevent this situation, the random process is used.

In each iteration, all images will be presented to the system. However, if the sequence of presentation of nodes is fixed, the map will be influence heavily by the first few images. Therefore, the order of presentation is shuffled in the beginning of each iteration to minimize this effect. When an image i is presented to the system, the feature vector x_i is compared with all weight vector $w_j(t)$ of all nodes in Kohonen Layer. The most similar node, called the winning node, can be found. Similarity is defined by Euclidean distance. Denoting the winning node as $c_i(t)$, finding a winning node is formulated as:

$$c_i(t) = \arg\min_{j \in M} \left\| x_i - w_j(t) \right\| \tag{3-8}$$

Then the winning node $c_i(t)$ and its neighbors are updated. The objective of updating is to make the weights of the winning node and its neighbors become more similar to that input vector. The formula and the illustration of the vectors are shown as follow.

$$w_i(t+1) = w_i(t) + \eta h_{ci}(t)[x - w_i(t)] \tag{3-9}$$

where $0 < \eta < 1$ is the learning factor and $h_{ci}(t)$ is the neighborhood function.

For convergence it is necessary $\eta h_{ci}(t) \to 0$ when $t \to \infty$. In this research, the learning factor η is a linearly decreasing function, of which the initial and final values are specified explicitly.

In this research, a constant weighting for the neighborhood function is used. The area

of it diminished in respect to time.

$$h_{ci}(t)=\begin{cases}1 & \text{if } i \in N_c(t)\\ 0 & \text{if } i \notin N_c(t)\end{cases} \tag{3-10}$$

where $N_c(t)$ refers to a neighborhood set of array points around node c at time t. It has been depicted as:

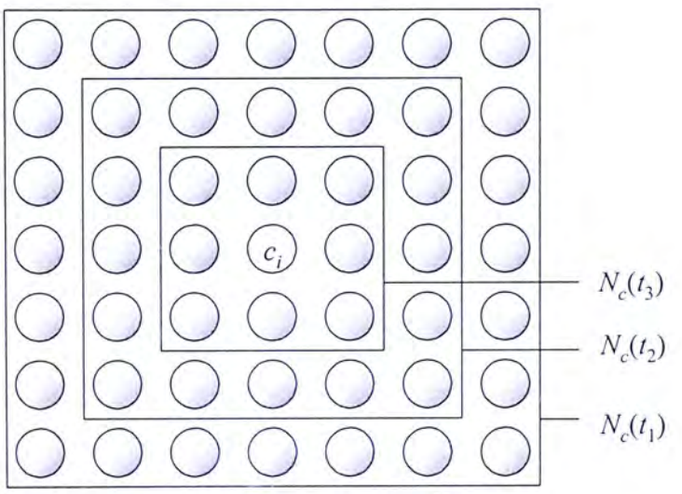


Figure 3-4. Neighborhood set ($t_1 < t_2 < t_3$)

The length of the side of square shape neighborhood set is defined as:

$$length(t)=\max\left(\max\left(SOM_{width},SOM_{height}\right)(1-t)^2,3\right) \tag{3-11}$$

At the beginning (t = 0), the neighborhood set covers at most the whole SOM. Its size diminished with respect to t until the length equals to 3.

After the network is trained through a number of iterations, a converged SOM is generated. The resultant SOM contains weights which represent the distribution of input space. Finally, feature vectors of images are mapped to the nodes by the minimum Euclidean distance, formulated as:

$$k_i = \arg\min_{j \in M} \left\| x_i - w_j(t) \right\|$$
$$R_j = \left\{ i \mid k_i = j \right\} \tag{3-12}$$

where k_i denotes the node which image i is mapped to,

and R_j denotes the set of images mapped to node j.

Labeling: After mapping the feature vectors, nodes in the SOM may contain zero or many images. To visualize the SOM, a representative image of each node is used as label. Three algorithms of labeling are tested in this research as follows.

1. Label by *similarity*: For each node j in the map, find an image $i \in I$ as label such that the Euclidean distance between x_i and w_j is minimal.

$$v_j = \arg\min_{i \in I} \left\| x_i - w_j \right\| \tag{3-13}$$

2. Label by *result-mean*: For each node j in the map, calculate the mean y_j of weight vectors of images in R_j. Next, find the image with minimal Euclidean distance to the mean as label.

$$y_j = \frac{\sum_{k \in R_j} x_k}{\left| R_j \right|}$$
$$v_j = \begin{cases} \arg\min_{i \in R_j} \left\| x_i - y_j \right\| & \text{if } \left| R_j \right| > 0 \\ \text{undefined} & \text{otherwise} \end{cases} \tag{3-14}$$

3. Label by *result-similarity*: For each node j in the map, find an image $i \in R_j$ such

that the Euclidean distance between w_j and x_i is minimal.

$$v_j = \begin{cases} \arg\min_{i \in R_j} \|x_i - w_j\| & \text{if } |R_j| > 0 \\ \text{undefined} & \text{otherwise} \end{cases} \quad (3\text{-}15)$$

3.2. Preliminary Experiment

This section offers some results of preliminary experiment testing the performance of this approach. The system is implemented in Java (JDK 1.1.8) and run on a Pentium II-233 PC. There are two modules in the system. The first one is feature module, which extracts chromatic and textural feature of images in the database. It also includes user-interface for retrieving feature information of each image. A simple query-by-example is implemented in this module for the purpose of testing the performance of the features. The second one is SOM module which trains SOMs using feature databases. It contains a user-interface to visualize the labeled result for browsing the images. In the following, SOMs using different image features and labeling method will be evaluated.

3.2.1. Feature differences

A database with 500 color images (640x480x24bits) have been used in the following experiments. Figure 3-5 (a-c) shows a 10x10 SOM trained with different features. Three SOMs are trained for 50 iterations[1]. Learning rate begins at 0.5 and ends at 0.01. Resulting maps are labeled by result-similarity. The number in left-lower corner in each node indicates the number of images in the node ($|R_j|$). After selecting a node in the map by user, the right column displays the images (R_j) mapped to the selected

[1] The term "iteration" in our algorithm is defined as number of presentation of all images (see section 2.2). However, most SOM algorithms define "iteration" as number of presentation of input patterns. Using the latter definition, there are 500 x 50 = 25000 iterations in this experiment

node.

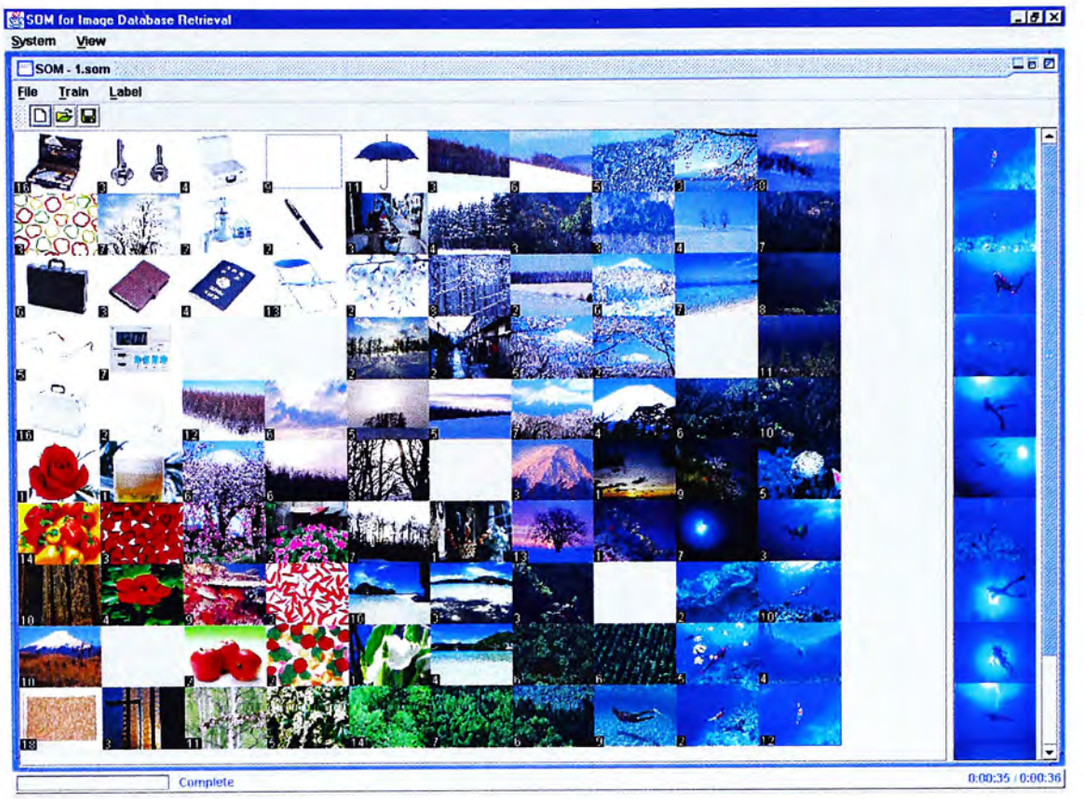
All the resulting maps can provide an overview of the database. And more importantly, they contain rich information. They visualize the underlying structure of the feature space by displaying topologically regions of major concepts in the image space and the distribution of the concept regions. The three SOMs will be evaluated in the following.

SOM in figure 3-5(a) only uses HS-histogram, a chromatic feature. It visualizes the distribution of color of images. For example, images with blue are put at the right-bottom of map. However, it cannot differentiate images with similar colors but different textures. For example, some winter scenery photos and some object photos both are almost black and white in color, therefore the concepts of these two categorizes of images are not clearly distinguished in the map. Also, objects with different colors like tennis-ball and baseball are separate apart.

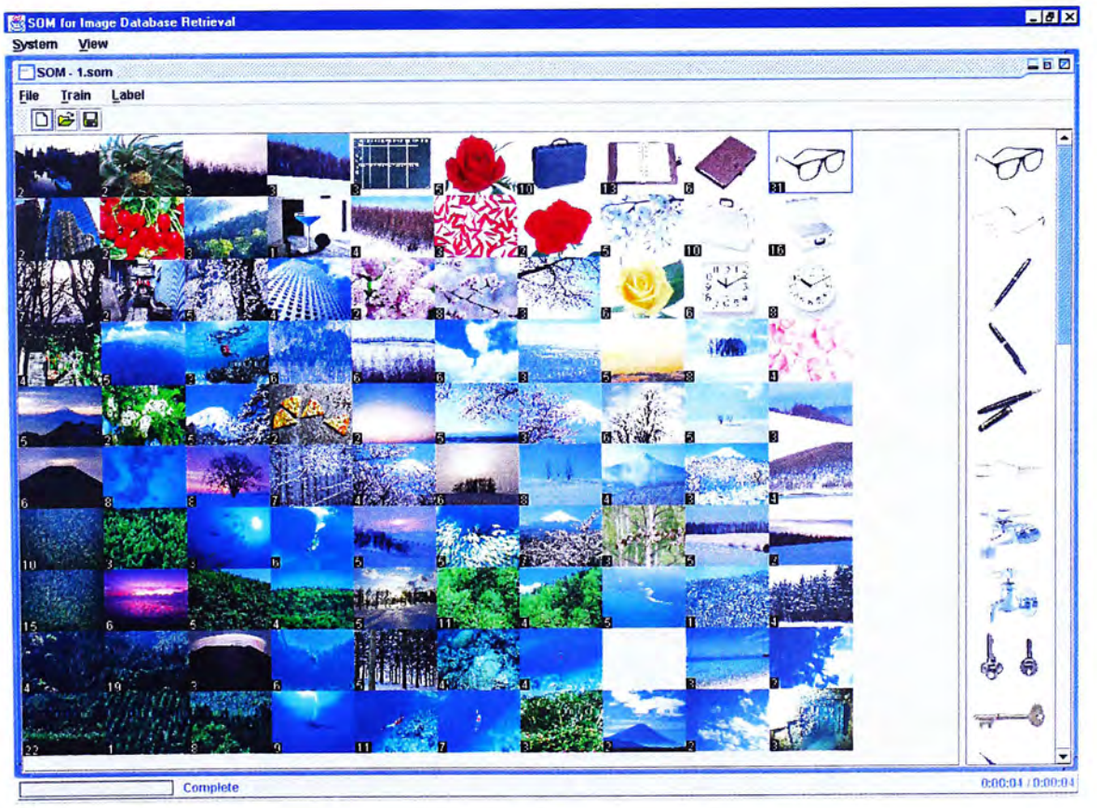
Gabor filter, a textural feature, is used to train the SOM shown in figure 3-5(b). Contrary to the former SOM, it can separate objects from scenery photos perfectly but photos with similar colors are not grouped together. For instance, photos with blue and green are in different regions in the map.

The final map combines HS-histogram and Gabor filter. The advantages of chromatic and textural features are found in this map. Objects and scenery photos are separate apart and images with similar colors are put together. Of course the performance is not as good as figure 3-5(a) in terms of chromatic differentiate performance, and figure 3-5(b) in terms of textural differentiate performance. Nevertheless, it provides

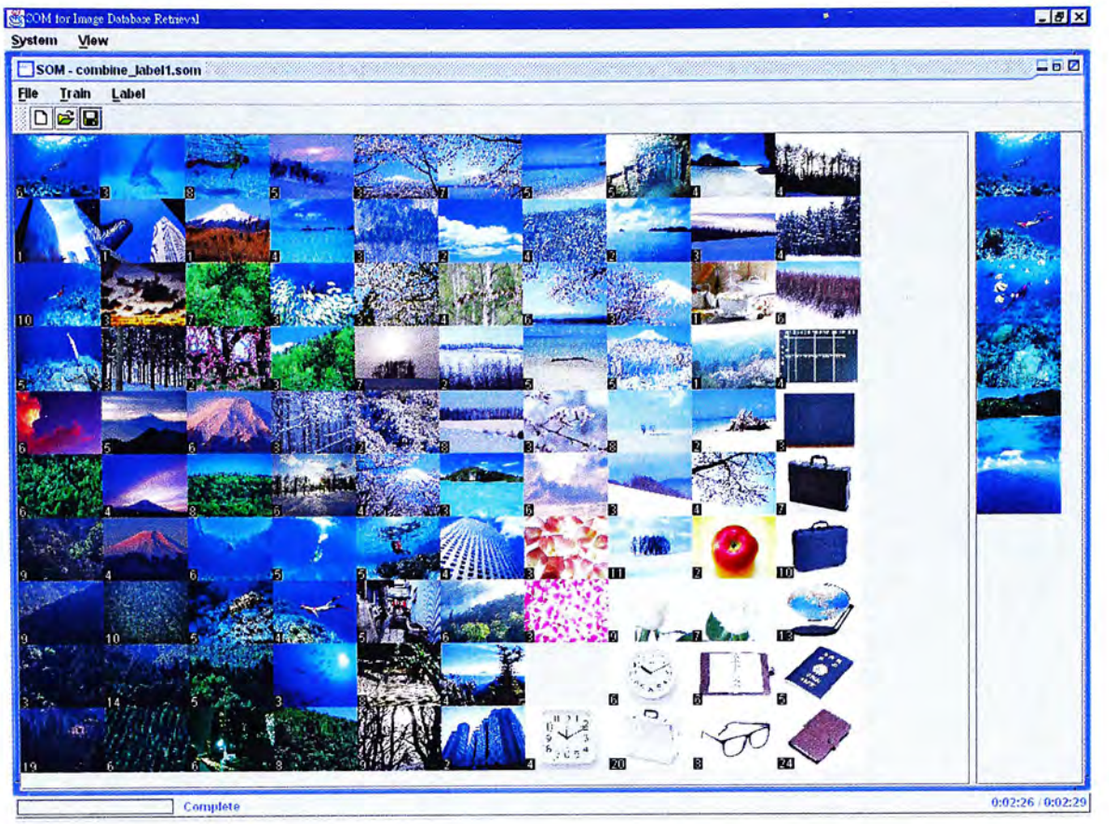
a balance of using both features.



3-5(a) HS-Histogram



3-5(b) Gabor Filter



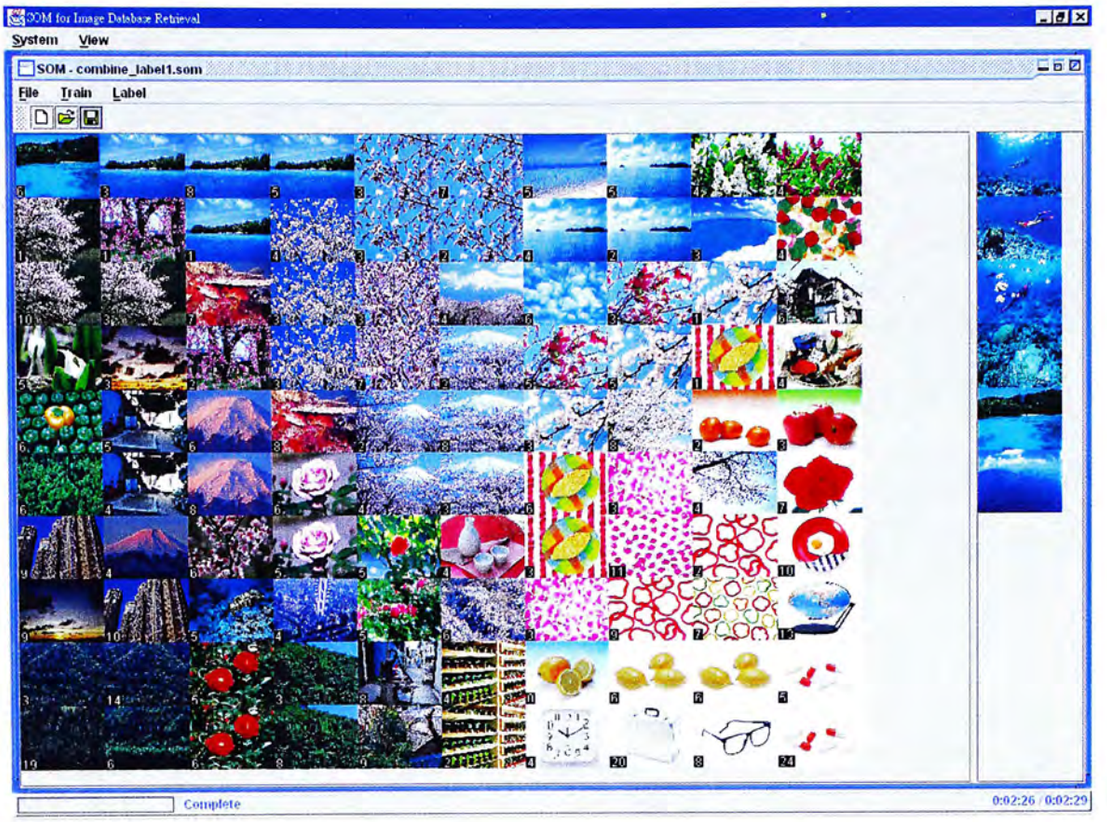
3-5(c) Combination of HS-histogram and Gabor Filter

Figure 3-5(a-c): 10x10 SOMs trained by different features

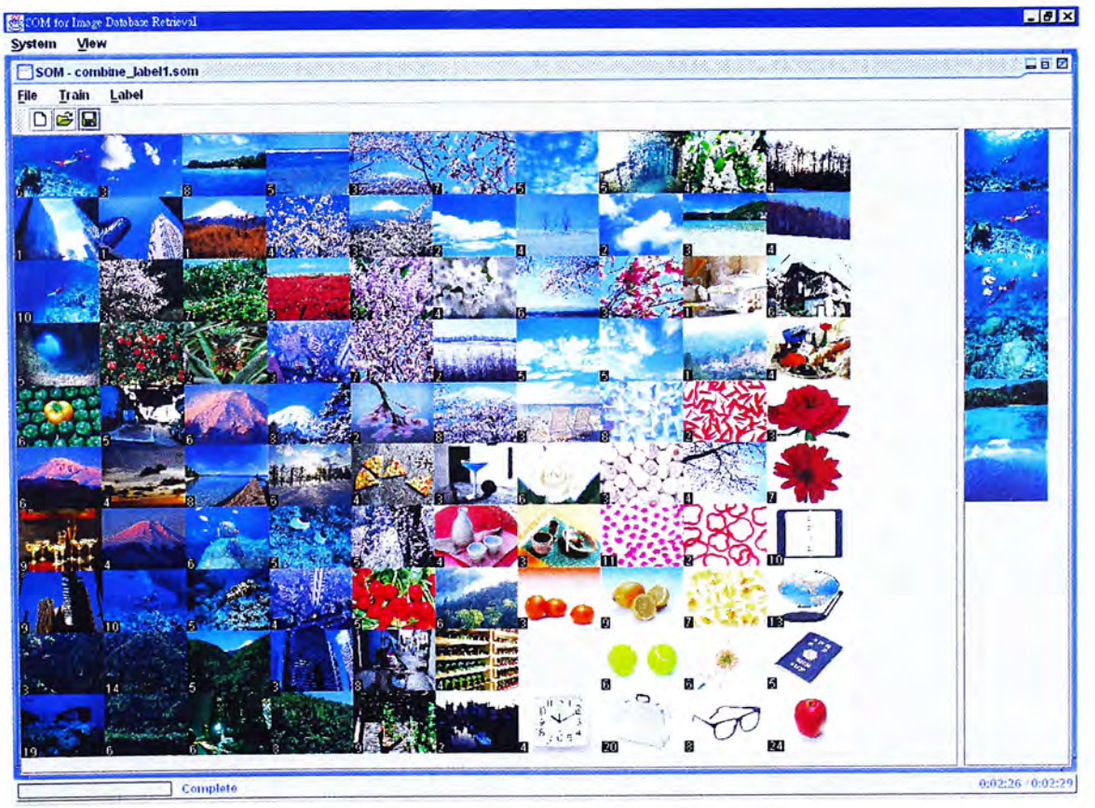
(all three maps are labeled by result-similarity)

3.2.2. Labeling differences

comparison has been made on different labeling methods. Figures 3-5(c) and 3-6(a-b) displays the same SOM trained by the same parameters in section 3.2.1 but labeled with different methods.



3-6(a) Label by Similarity



3-6(b) Label by Result-Mean

Figure 3-6(a-b): 10x10 SOMs labeled by different methods

(both are trained by combined features)

The map in figure 3-6(a) is labeled by similarity, that is, using the most similar image in the database to represent the node. It gives the best performance in visualizing the information of the weights trained in the map thus present an overview of distribution of images in the database and the map is generally very continuous. However, the label image of a node may not be an image mapped to that node. And there are cases of duplicate representative images. If it is used as a user-interface for browsing, users may be confused if their interested image exists repeatedly throughout the map; they simply do not know which node they should explore further.

Figure 3-6(b) shows a map labeled by result-mean. The label image is chosen from the most similar image to the mean of features vector of images mapped to that node. It prevents the just mentioned problem of labeling by similarity and it finds the best images to represent images mapped to that node. However, the label images are not the most similar to the weights of the nodes thus the map is visually more fragmented.

The last map is shown in figure 3-5(c), which is labeled by result-similarity. It has some advantages of labeling by similarity and labeling by result-mean. It provides a balance between representing weights of nodes and representing the features of images mapped to the nodes. Two important properties of result-mean and result-similarity methods are that the label image must be included in images mapped to the node and duplicate label image will not exist.

4. Experiment

An experiment was conducted for evaluating the performance between the visual thesaurus (**SOM**) and Query-By-Example (**QBE**) systems. In this experiment, Human evaluation has been used to measure the performance.

4.1. Subjects

There are 34 undergraduates of the department of System Engineering and Engineering Management participated in this experiment. 14 subjects are male. The mean age is 21.4. Subjects are divided into 2 groups, of which each has 17 subjects.

4.2. Apparatus

4.2.1. Systems

Each subject used one Pentium III PC with 15" CRT monitor. A Java application has been developed as prototypes of SOM and QBE systems.

4.2.2. Test Databases

In the experiment, three sets of images were used. The first set (**Trial**) is 200 images of flowers. The second set (**Cloth**) is 469 cloth textiles which are downloaded from web. The third set (**All**) is 2000 images of flower, forests, sky, underwater scenery, food and building from a stock image collection.

Since the effect of different features is also investigated, the third set (**All**) is subdivided into two databases using different features: hue-saturation histogram (**Allhs**) and combined-feature (**Allcf**). Combined-feature combines chromatic (hue-saturation histogram) and texture (Gabor filter) features.

As SOM contains stochastic training procedure which makes different output for the same input, 10 SOMs have been generated for the same database and randomly selected one for each subject, in order to minimize the effect of individual training.

4.3. Procedure

In this experiment, subjects are required to find images that match the target. There are two types of targets: **text** and **image**. For the text target, the application firstly shows a list of text description. And then the subject can choose and combine the text descriptions as a target feature. Then the subject needs to find one or multiple images that match the text description. For the image target, the application randomly picks an image from the database as target. Subject needs to find an identical image.

There are seven types of sections as follows. The detailed description is given afterwards.

Section	Description
Sign-up	For subject entering his/her name, student/staff ID, age and gender.
Description	For subject to choose 6 text description as target.
SOM (text)	Using SOM to find images relevant to 3 text targets that is defined by the subject.
SOM (image)	Using SOM to find images identical to 3 randomly generated image targets.
QBE (text)	Using QBE to find images relevant to 3 text targets that is defined by the subject.
QBE (image)	Using QBE to find images identical to 3 randomly generated image targets.
Questionnaire	For subject filling in a form of questionnaire.

4.3.1. Description

The feature dialog is shown in figure 4-1. Some features of the database are listed in a tree-structure control. For example, in the trial database, the first level is the number of flowers in the image and the second level is the major color of flower(s). Subject needs to choose a combination of the features and optionally enters an additional text description. This section requires the subjects to choose 6 descriptions before using the two systems.

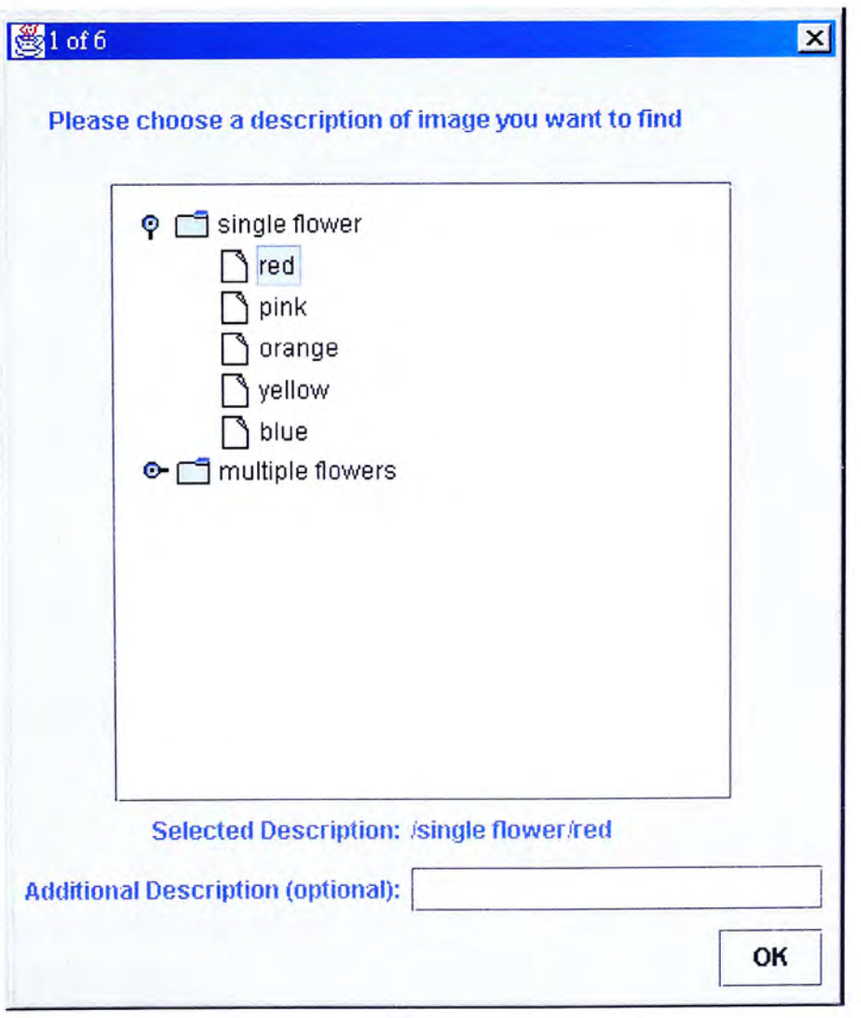


Figure 4-1: Feature Dialog

4.3.2. SOM (text)

In this section, the subject is required to find images relevant to the text description which he/she has chosen. The SOM (text) environment is displayed as follows.

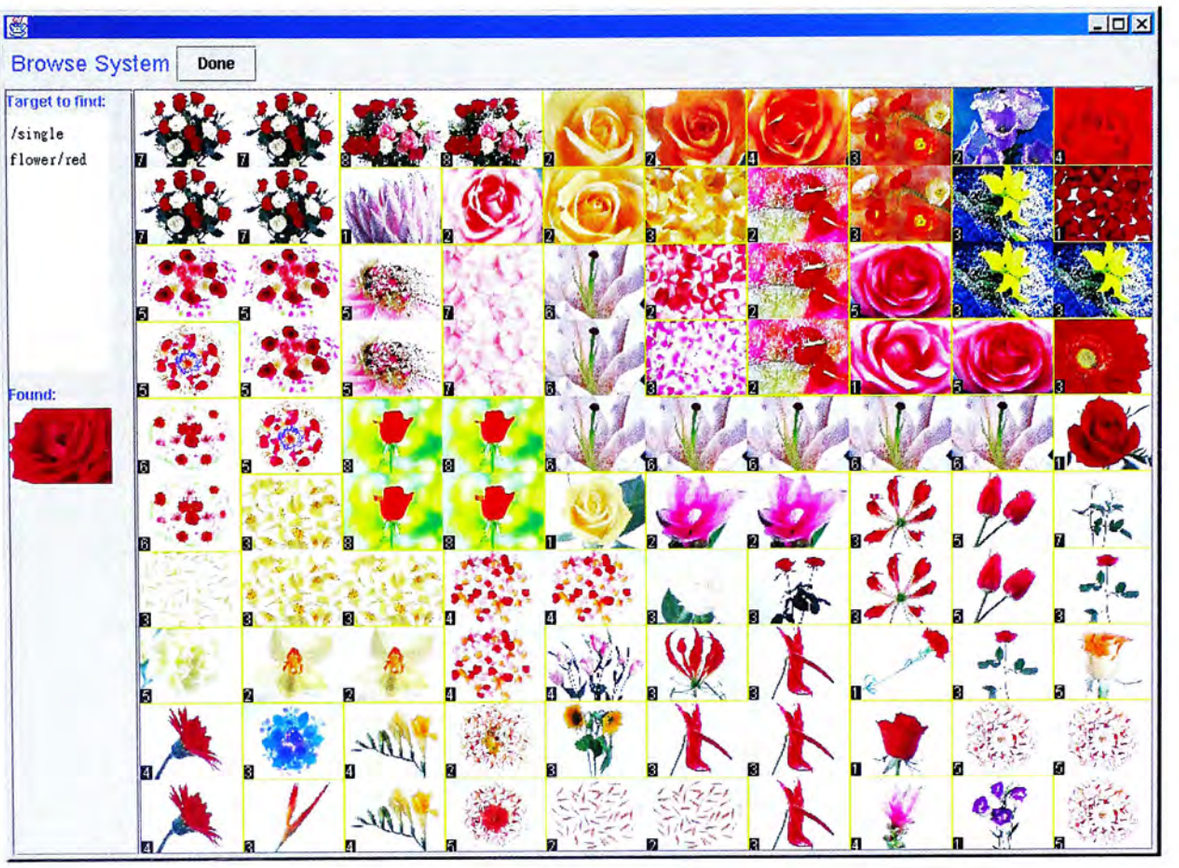
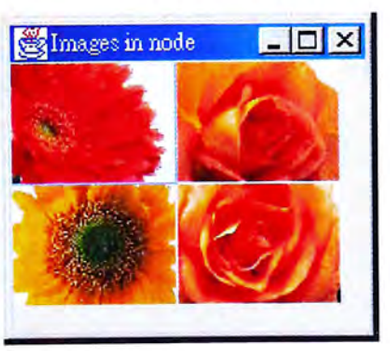


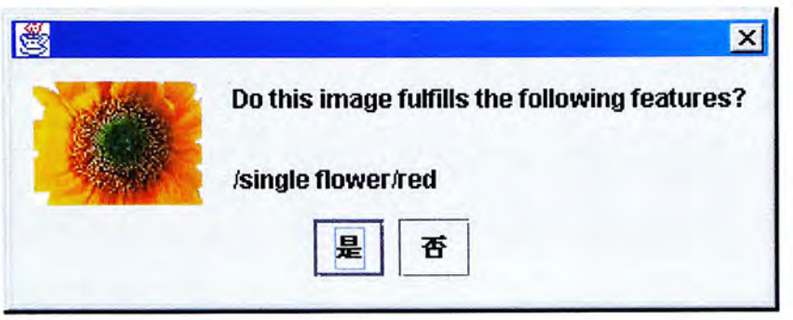
Figure 4-2: SOM (text) environment

- The left-top part (feature panel) shows the text features chosen by the subject.
- The left-bottom part (found panel) shows the images chosen as relevant by the subject.
- The right part (SOM panel) shows the image labels of SOM.

When the subject clicks on a node in SOM panel, the result panel shows images associated with the node.



If the subject finds that one of the images in the result panel is relevant, the subject needs to click on the image. A confirmation dialog is shown and the subject needs to click "yes" to proceed or "no" to continue to try on the other images.



When the subject thinks that no more images are relevant, he/she can click "done" to complete the task. There is a time limit for the whole section (see Table 4-1).

4.3.3. SOM (image)

In this section, subject is required to find an image in SOM identical to an image target (key) that is generated randomly by the system once in each trial. The SOM (image) environment is displayed as follows.

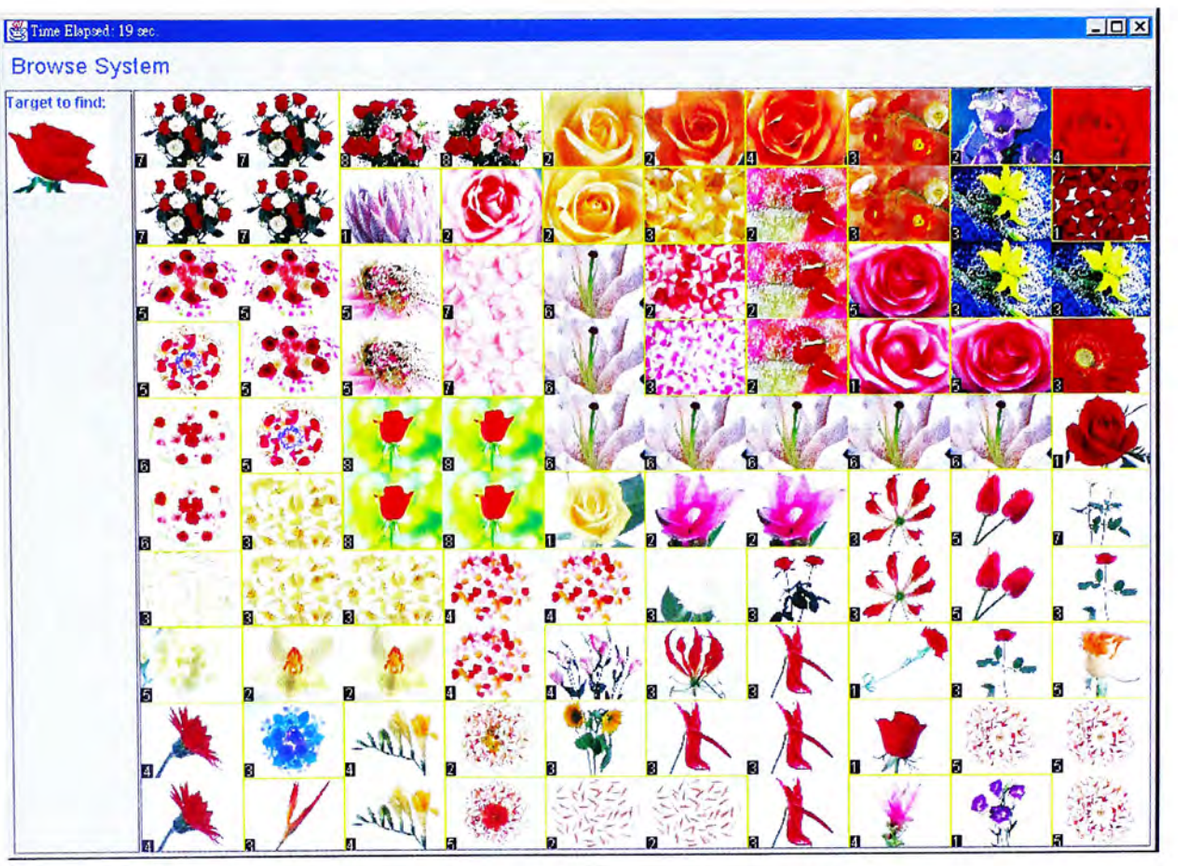
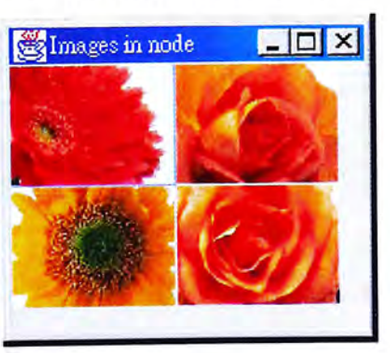


Figure 4-3: SOM (image) enviornment

- The left-top part (target panel) shows one image target at a time.
- The right part (SOM panel) shows the image labels of SOM.

When the subject clicks on a node in SOM panel, a window pop-up to show images associated with the node.



If the subject finds that one of the images in the result panel is identical to the target image, the subject needs to click on the image. If the system determines that the image is identical, a dialog is shown. Subject can click "ok" to proceed. There is a time limit for each trial. (see Table 4-1)

4.3.4. QBE (text)

The QBE environment is shown as follows.

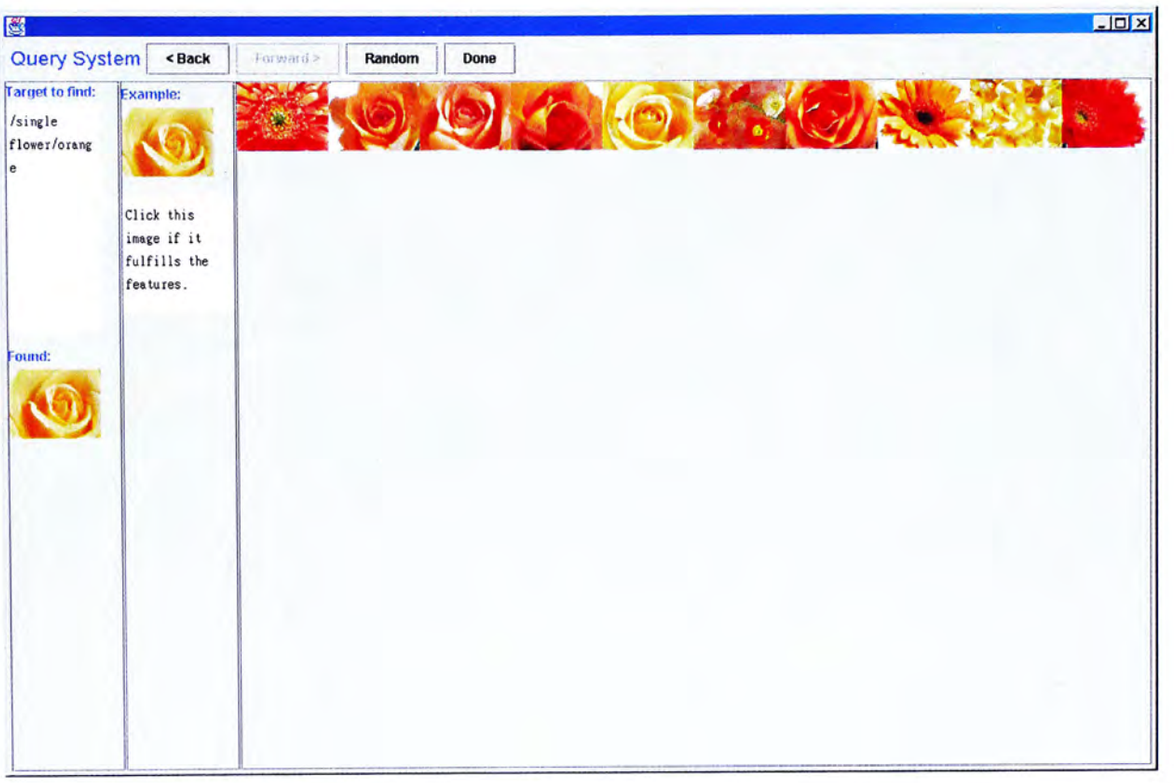


Figure 4-4: QBE (text) environment

- The left part (feature panel) shows the text features chosen by the subject.
- The center part (QBE panel) shows the images of search result. The order of ranking is from top to down, left to right. The first rank is omitted because it is shown in example.
- The right part (example panel) shows the current example image.
- The top part (navigate panel) shows paging buttons, current page number and total number of pages.

The QBE panel initially shows a set of random images. When the subject clicks a most relevant image in the QBE panel, it will become an example to query the database again. The example will be shown in the example panel. The number of result images shown in the QBE panel is equal to the mean of result images in SOM, in order to make a fair comparison.

This process continues until the subject clicks the example image in the example panel, which means the example is relevant to the features. Then, a confirmation dialog is shown and the subject need to click "yes" to proceed or "no" to continue to try on the other images. When the subject thinks that no more images are relevant, he/she can click "done" to complete the task. There is a time limit for the whole section.

4.3.5. QBE (image)

In this section, subject is required to find an image identical to an image target (key) that is generated randomly by the system once in each trial. The QBE (image) environment is shown as follows.

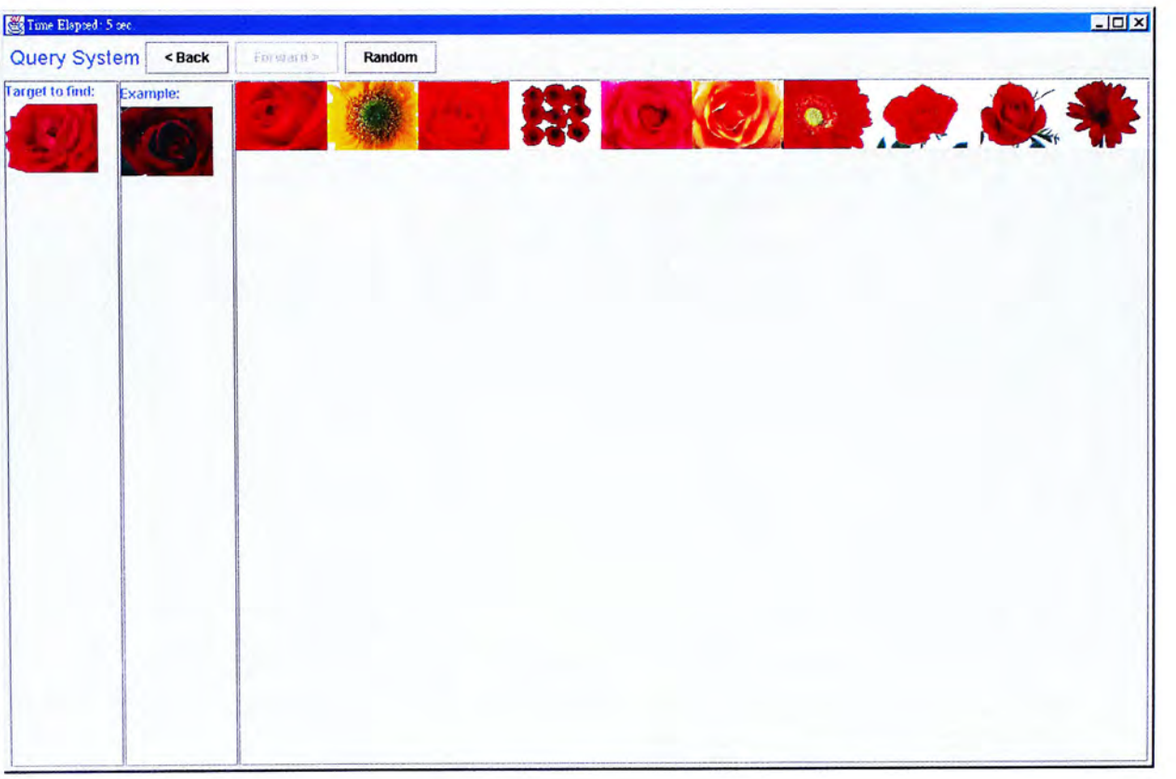


Figure 4-5: QBE (image) environment

- The left part (target panel) shows one image target at a time.
- The center part (example panel) shows the current example image.
- The right part (QBE panel) shows the images of search result. The order of ranking is shown in the figure. (the first rank is omitted because it is shown in example)
- The top part (navigate panel) shows navigation buttons.

The QBE panel initially shows a set of random images. When the subject clicks the most relevant image in the QBE panel, it will become an example to query the database again. The example will be shown in the example panel. The number of

result images shown in the QBE panel is equal to the mean of result images in SOM, in order to make a fair comparison.

This process continues until the subject clicks on an image that is identical to the target. If so, a dialog is shown. Subject can click "ok" to proceed. There is a time limit for each trial.

4.3.6. Questionnaire

the questionnaire is designed according to Doll et al.'s research on end-user computing satisfaction [Doll88]. The main purpose of the questionnaire is to compare the end-user satisfaction between two systems.

The application shows the same set of questions twice, one for SOM and another for QBE. In order to help user to distinguish the two systems, the application shows a screen-shot of the system. The dialog is shown as follows.

Figure 4-6: Questionnaire Dialog

4.3.7. Experiment Flow

In order to minimize the bias introduced by the order of systems, two subject groups have a different flow of sections. Group 1 will use SOM first and then QBE. Group 2 will use QBE first and then SOM. The flows are shown as follows:

Table 4-1: Experiment Flow

Step	Database	Time Limit (sec)	Group 1	Group 2
1	-	-	Sign-up	Sign-up
2	-	-	Description	Description
3	Trial	60	SOM (image)	QBE (image)
4	Trial	60	QBE (image)	SOM (image)
5	Trial	60	SOM (text)	QBE (text)
6	Trial	60	QBE (text)	SOM (text)
7	-	-	Description	Description
8	Cloth	60	SOM (image)	QBE (image)
9	Cloth	60	QBE (image)	SOM (image)
10	Cloth	60	SOM (text)	QBE (text)
11	Cloth	60	QBE (text)	SOM (text)
12	-	-	Description	Description
13	Allhs	180	SOM (image)	QBE (image)
14	Allhs	180	QBE (image)	SOM (image)
15	Allhs	180	SOM (text)	QBE (text)
16	Allhs	180	QBE (text)	SOM (text)
17	-	-	Description	Description
18	Allcf	180	SOM (image)	QBE (image)
19	Allcf	180	QBE (image)	SOM (image)
20	Allcf	180	SOM (text)	QBE (text)
21	Allcf	180	QBE (text)	SOM (text)
22	-	-	Questionnaire (SOM)	Questionnaire (QBE)
23	-	-	Questionnaire (QBE)	Questionnaire (SOM)

4.4. Results

The statistics of the two systems have been compared in term of number of success, time, and other measurements. The results have been grouped into text target and image target tasks, which is tabulated into the following two tables.

Table 4-2: Results of text target tasks

database	Cloth			Allhs			Allcf		
system	SOM	QBE	p-value	SOM	QBE	p-value	SOM	QBE	p-value
mean(no. of success)	4.76	4.33	0.4565	21.89	12.05	**0.0028**	11.73	9.52	0.1977
std(no. of success)	4.68	3.44		29.97	12.68		12.93	11.27	
mean(mean(success time))	29540.16	20612.03	0.0681	18431.81	20107.47	0.6792	15272.62	18940.01	0.2190
std(mean(success time))	43526.52	22110.16		33466.57	23084.30		17199.04	24434.58	
mean(no. of query)	46.43	29.05	**0.0003**	44.10	48.90	0.3385	39.31	40.00	0.8583
std(no. of query)	40.57	24.20		32.97	38.01		27.45	27.12	
mean(no. of unique query)	35.67	17.82	**0.0000**	31.71	28.49	0.2712	29.96	23.95	**0.0135**
std(no. of unique query)	29.19	16.26		21.43	19.97		18.33	15.86	
mean(query time)	2649.46	3379.28	**0.0017**	3431.19	3438.49	0.9736	2789.82	2634.76	0.3442
std(query time)	1773.90	1479.02		1715.80	1392.18		1332.25	961.65	

p-value is calculated by 2-tails t-test, assuming unequal variances

Table 4-3: Results of image target tasks

Database	Cloth			Allhs			Allcf		
System	SOM	QBE	p-value*	SOM	QBE	p-value*	SOM	QBE	p-value*
No. of success	65	60	0.500	70	55	0.2616	73	72	0.500
success rate	64 %	59%		69%	54%		72%	71%	
mean(task time)	36266.60	37220.74	0.7661	106004.53	118348.35	0.1708	90791.28	84001.31	0.4867
std(task time)	22772.80	22983.41		62768.77	65467.45		68335.73	70803.74	
mean(no. of query)	6.77	7.03	0.8552	18.53	14.07	0.0857	16.33	10.65	**0.0204**
std(no. of query)	7.42	8.50		16.51	12.00		17.01	11.37	
mean(no. of unique query)	5.74	4.70	0.2899	13.99	10.35	**0.0425**	13.00	7.78	**0.0029**
std(no. of unique query)	6.01	4.80		11.79	7.84		12.37	7.59	
mean(mean(query time))	3436.43	3254.62	0.5674	4724.07	5287.16	0.2810	3856.16	4097.51	0.4375
std(mean(query time))	1844.51	1673.52		2408.90	3166.92		1540.11	2116.37	

*p-value in no. of success is calculated by 1-tail paired t-test; others sections are calculated by 2-tails t-test, assuming unequal variances.

Table 4-4: Score of answers in questionnaire

Answers for Questions 1 - 19	Answers for Question 20	Score
Almost never	Non-existent	0
Some of the time	Poor	1
About half of the time	Fair	2
Most of the time	Good	3
Almost always	Excellent	4

Table 4-5: Results of questionnaire

Questions	Factor	Mean(diff)	Std(diff)	p-value
1. Is it easy to correct errors?	Ease	**0.382353**	1.128547	**0.02831**
2. Do you enjoy using the system?	General	**0.088235**	1.190051	0.334157
3. Do you think the output is presented in a useful format?	Format	**0.058824**	1.179141	0.386479
4. Is the system difficult to operate?	-Ease	**-0.05882**	1.347077	0.400297
5. Are you happy with the layout of the output?	Format	-0.14706	1.076818	0.215769
6. Is the system accurate?	Accuracy	**0.147059**	1.104601	0.221553
7. Do you get the information you need in time?	Timeliness	**0.088235**	1.083419	0.319
8. Do you find the output relevant?	Format	0	1.015038	0.5
9. Is the system easy to use?	Ease	-0.02941	0.869876	0.422459
10. Is the system user friendly?	Ease	-0.02941	1.086705	0.437782
11. Is the system efficient?	Timeliness	-0.02941	1.290649	0.447549
12. Is the system troublesome?	-General	**-0.08824**	1.137985	0.327073
13. Is the system convenient?	Timeliness	-0.05882	1.204566	0.388809
14. Is the system difficult to interact with?	Ease	0.029412	1.167367	0.442048
15. Does the system provide comprehensive information?	Content	**0.205882**	0.880062	0.090885
16. Would you like more concise output?	- Accuracy	0.294118	0.798841	**0.019624**
17. Would you like the system to be modified or redesigned?	General	-0.14706	0.95766	0.188528
18. Are you satisfied with the system?	General	**0.205882**	0.913847	0.099009
19. Do you get information fast enough?	Timeliness	**0.058824**	1.153156	0.383997
20. Overall, how would you rate your satisfaction?		**0.176471**	0.833779	0.112936

- Diff is calculated as Score(SOM) – Score(QBE)
- p-value is calculated by 1-tail paired t-test
- Bold Mean(Diff) indicates that SOM out-perform than QBE.
- Bold p-value highlights p-value < 0.05

Table 4-6: Correlation of Diff, with bold values indicate significant at 95% level of confidence

2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
-0.0033	0.0509	-0.3037	0.0227	0.2209	0.2938	0.2645	0.1044	0.1330	0.1744	-0.0437	0.2400	-0.0778	0.2234	-0.0277	0.0256	0.1564	0.0288	0.1516	1
	0.5145	-0.1857	**0.3888**	**0.6122**	**0.6049**	**0.4265**	0.2660	0.2833	**0.4950**	-0.1955	**0.4688**	0.1944	0.1847	0.0994	**-0.3605**	**0.5958**	**0.6144**	**0.7779**	2
		-0.1695	**0.5798**	**0.5981**	**0.4939**	**0.5064**	0.3563	0.3325	**0.3994**	-0.0638	0.3225	-0.1334	0.0172	**0.3350**	-0.2605	**0.5790**	**0.5991**	**0.6056**	3
			0.0147	-0.1366	-0.2455	-0.1995	0.1795	-0.0633	-0.0882	-0.1221	-0.1703	0.0204	**-0.3729**	-0.0679	-0.2418	-0.0391	0.0998	-0.2063	4
				0.3754	**0.4271**	**0.4990**	**0.4482**	**0.4882**	0.2584	-0.1593	0.2735	-0.1411	0.2887	0.2279	-0.0510	**0.4320**	**0.5197**	**0.6036**	5
					0.4953	**0.4054**	**0.3515**	0.1552	**0.4707**	-0.0858	**0.4394**	0.0670	-0.0009	0.1555	-0.2368	**0.5395**	**0.6115**	**0.4974**	6
						0.4684	0.3244	**0.4141**	**0.5437**	-0.0672	**0.5382**	0.0458	0.3300	0.0391	-0.0747	**0.3484**	**0.5051**	**0.6532**	7
							0.5491	0.3022	0.1388	-0.0262	0.1983	-0.2557	**0.4410**	0.1495	-0.1247	0.2613	**0.4401**	**0.5013**	8
								0.5761	0.1342	-0.2170	0.3164	-0.2677	0.2852	**0.4053**	-0.0781	0.2747	**0.4549**	0.2998	9
									0.2154	0.0223	0.2764	-0.2143	**0.3551**	0.1848	0.1704	0.2504	0.2916	0.4072	10
										-0.4145	**0.7980**	-0.1000	0.0855	0.0380	-0.2243	**0.4934**	**0.4695**	**0.5682**	11
											-0.5566	0.3214	0.0187	-0.1373	0.0990	0.0180	**-0.3885**	-0.1428	12
												-0.2142	0.2690	0.1760	-0.0865	**0.4243**	**0.5480**	**0.5537**	13
													0.0824	-0.2045	-0.1044	0.1930	-0.3165	0.2124	14
														-0.0456	0.2527	0.0211	0.0176	0.2794	15
															0.2167	0.2881	0.1780	0.1017	16
																-0.4145	-0.3212	-0.2322	17
																	0.4483	**0.6270**	18
																		0.4616	19

4.5. Discussion

Based on the experiment results presented in Table 4-2, comparison has been made with the performance of text target tasks using the two systems. Some interesting observations have been made, analyzed and discussison is as follows.

(1) Using SOM, subjects could find 38% more images that are relevant to the text description target on average, especially significant in the **Allhs** database ($p = 0.0028$). The performance of SOM is higher than QBE in this task.

(2) There is no significant difference in the time required to find a relevant image between SOM and QBE. Combining this observation and (1), the results show that SOM is empirically more efficient than QBE in this task.

(3) Using SOM, subjects generally submitted more queries (on average 16% more for all queries, and 46% more for unique queries). There are three out of six cases that are very significant ($p = 0.0003$, $p = 0.0000$, $p = 0.0135$). This observation shows that subjects were encouraged to generate more queries using SOM. Also, using QBE, subjects generated 47% more duplicate queries on average.

For the finding identical image task, Table 4-3 compares the performance between the two systems. Similar observation can be drawn as follows.

(1) The success rate of SOM is higher than QBE by 12% on average. The performance generally is higher but the results are not numerically significant.

(2) There is no significant difference in time between SOM and QBE. This finding is also found in observation (2) in text target task.

(3) Using SOM, subjects generally submitted more queries (on average 27% more for all queries, and 41% more for unique queries). There are three out of six cases that are significant ($p = 0.0204$, $p = 0.0425$, $p = 0.0029$).

Subjects require approximately the same time for performing these two types of tasks. However, due to the higher performance by using SOM, SOM has a higher efficiency. The reason for this is that QBE on average generates less possible queries than SOM. In addition, subjects can remember their historical queries in SOM because the possible queries are fixed spatially in the user interface. Consequently, the performance is highly related to the number of queries and the quality of queries. In SOM, spatially near labels on the map are similar images. This essential property of SOM can help the user refine their query easily. On the other hand, the limited possible examples in QBE may cause user being confined to a looping set of small number of images as query.

The performance of the experiment varies for different feature set and database. In the experiment, Allhs database yields the best performance. This may be related to the subjects' lack of understanding of the underlying mechanism of texture features. Combined feature may confuse subjects under the conditions that the returned images are relevant in both colors and textures. When the similarity is not very high among the result sets, use of combined features may retrieve a set of images which are not visually similar to each other. This can be explained by the similarity can be dominated by either color or texture only. For general color images with much diversified color properties, similar to the one used in Allhs and Allcf, using only chromatic feature may be a better choice.

These empirical findings support fact that SOM can provide higher efficiency than the traditional QBE approach.

The questionnaire is used to reflect the user satisfaction between two systems. Results from table 4-5 and 4-6 are analyzed as follows.

(1) In overall (Q20), subjects are more satisfied with SOM ($p = 0.1130$). The satisfaction is correlated to almost all questions. (10 out of 19 are significant)

(2) Question 1 and 16 shows significant evidence that from the user's point of view, SOM is easier to correct errors but the output is less accurate than QBE. This result is compatible with the fact that users submit more queries in SOM than in QBE. The fact that SOM is less accurate in this sense can be justified by the single assignment policy, which means one image is assigned to one node only. Thus, user can find a specific image in one node only.

(3) Question 14 shall be omitted due to there being no significant correlations with other 19 questions. This may be due to a misunderstanding of the question.

After eliminatination question 14 and grouping of the results into factors, the score of components are shown as follows.

Table 4-7: End user satisfaction in separate components

Accuracy	-0.073530
Format	0.088235
Ease of Use	0.095588
Timeliness	0.014707
General	0.058824

Table 4-7 illustrates some important issues about the user satisfaction:

(1) SOM is less accurate then QBE. This is consistent with the previous analysis.

(2) Subjects prefer the format of SOM than QBE because SOM presents the output in a more useful format and it provides more comprehensive results. This can be

attributed to the natures of SOM, which provides an overview of the whole database and the feature similarity relationship between neighbor nodes.

(3) Although there are results from a few questions indicating that SOM is not easy to use, SOM can allow users to easily correct errors. On average SOM gets a positive score on ease of use.

(4) Subjects think that by using SOM they can get information faster than using QBE. However, the empirical results show that there is no significant difference in terms of time. That subjects perceive better timeliness in SOM may be explained by other factors including accuracy and ease of use.

(5) Generally, subjects are more satisfied with SOM than QBE, as shown with the general component and overall comment (Q20).

In conclusion, the empirical results and the survey in this experiment have generally supported the hypothesis that SOM has a higher level of performance and user satisfaction.

5. Quantizing Color Histogram

Recently there have been numerous projects using the color histogram as one of the visual features, different color coordinate systems. For example, RGB, YIQ, CIELUV are among those that have been employed. There are two major approaches to quantize the color space. With the first, each axis of color coordinates is quantized uniformly to generate a fixed number of bins. In the second, the color space is arbitrarily quantized.

The first approach has been applied in many CBIR systems [Flickner95, Swain91, Pentland94]. However, it appears to be most effective when applied only to RGB only. For other non-cubical color coordinate systems, some bins will be always empty. When those empty bins from the histogram are dropped to reduce dimension, the bins in the boundary of the color space cannot be fully utilized. Consequently, this process results in reduced performance.

Alternatively, some researches have applied the second approach. For example, Smith et. al [Smith96] have quantized the HSV space uniformly to 18 hues, 3 saturations, 3 values and 4 grays. There are 166 quantized colors. However, there is not sufficiently strong evidence regarding this approach to indicate that the quantized color space is compact.

Nevertheless, for the first approach, RGB is not a perceptually uniform color space. Alternatively, the CIE (Commission Internationale de L'Éclairage) has standardized two perceptually uniform color spaces: CIELAB (CIE L*a*b*) and CIELUV (CIE L*u*v*). There is however, a problem with using these color spaces for color

histogram. These color spaces are not cubical, generating bins by quantizing each axis uniformly generate empty or partially used bins.

In the current research a new approach has been proposed [Yang02b] to tessellate non-cubical color space using a vector quantization (VQ) method, the General Loyld Algorithm (GLA) [Gersho92]. This approach also works on polar color spaces like HSV and LHS. Section 5.1 describes the algorithm in detail. For evaluation purpose, both objective indexes are used, namely indexing effectiveness and human evaluation as well as precision and recall to test the empirical performance of this innovative approach in section 5.2. Finally, conclusions and future works are presented in section 5.3.

5.1. Algorithm

An overview of the specific algorithm is shown in figure 5-1. The algorithm is divided in two phrases, codebook generation and histogram generation. In the codebook generation phrase, a color space is tessellated using GLA. This phrase is executed only once for each color coordinate system. In the histogram generation phrase, color histogram of each image is generated using the results from the previous phrase.

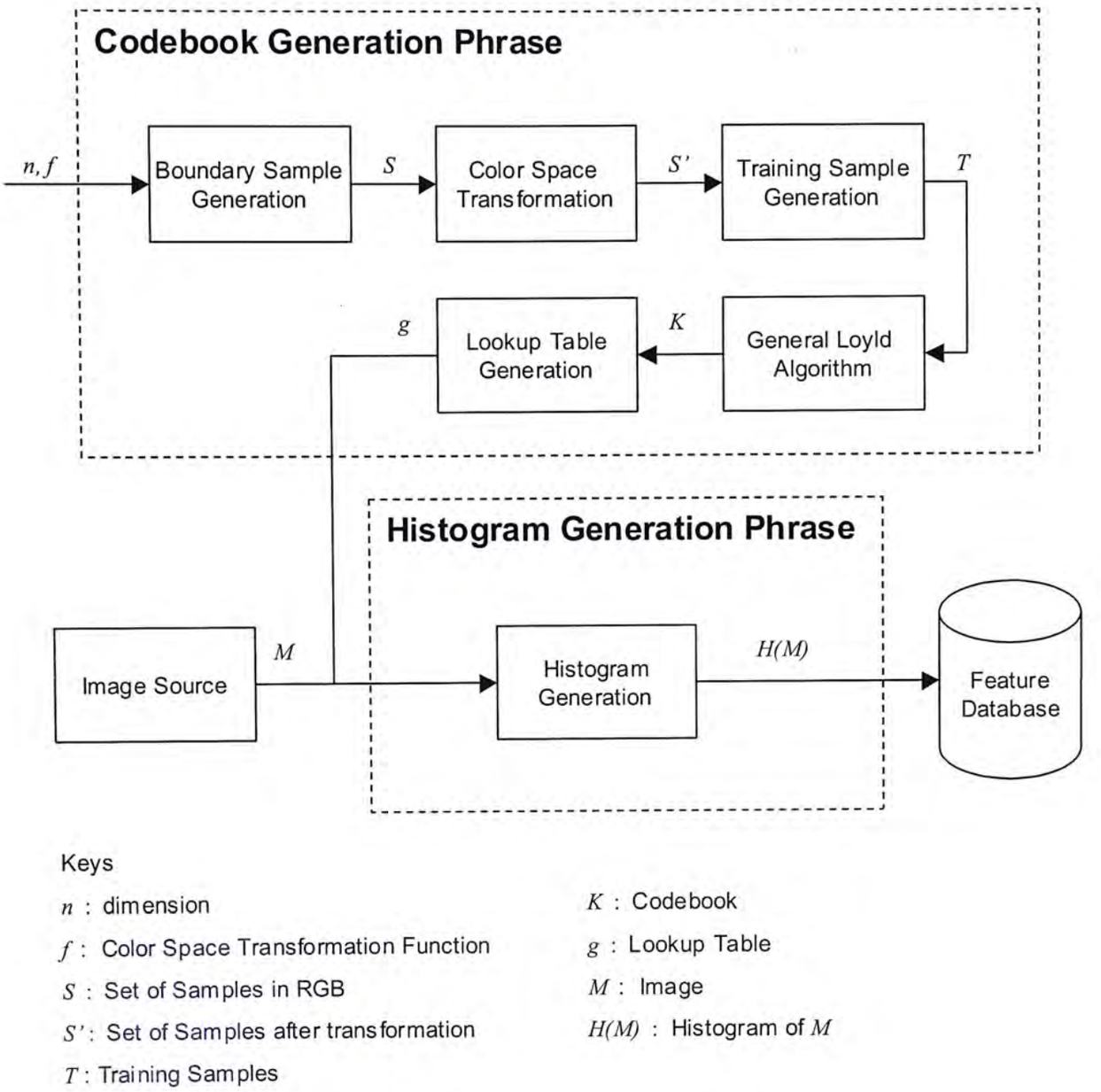


Figure 5-1: Overview of the algorithm

5.1.1. Codebook Generation Phrase

In this phrase, a lookup table is generated. The table maps from color space to histogram space. The procedure includes boundary samples generation, color space transformation, training sample generation, GLA and lookup table generation. To illustrate the procedure, Figure 5-2 illustrates the procedure for a two-dimension space.

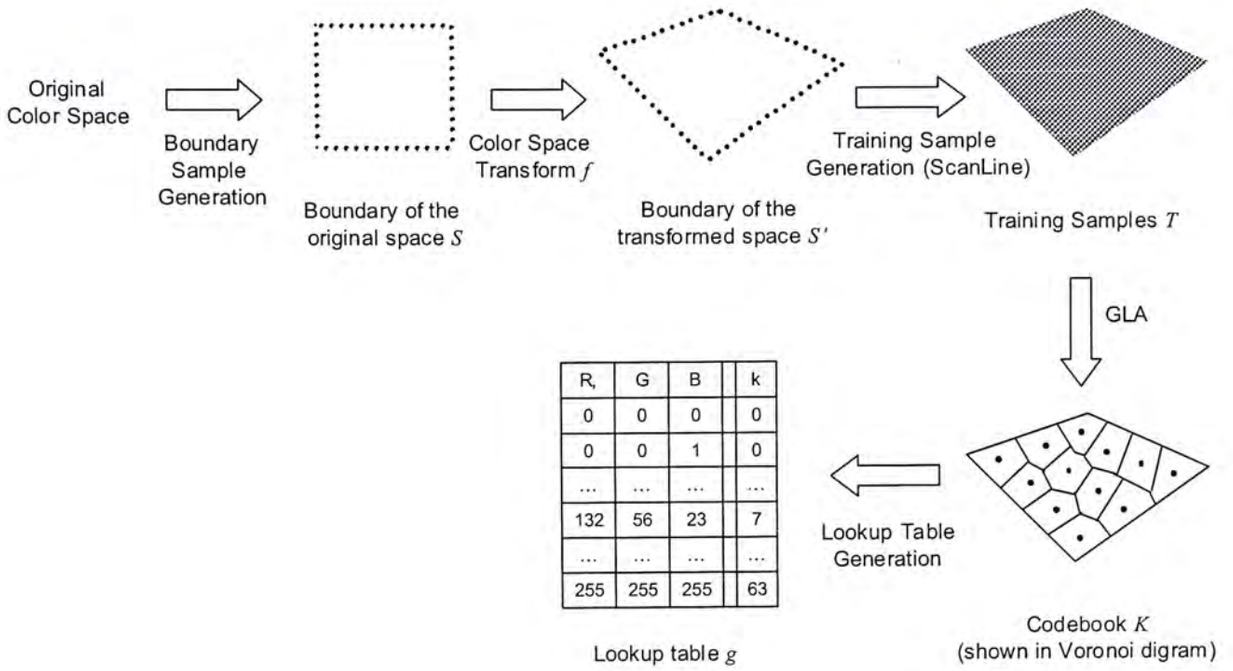


Figure 5-2: Codebook Generation (two-dimensional space)

First, the boundary of the original color space (the square in figure 5-2) is generated. In this algorithm, RGB has been utilized as the original color space, thus the boundary is comprised of six flat surfaces of the RGB cube. Subsequently this transforms the point samples of the boundary to a target color space by a color coordinate transformation function *f*, for example, RGB-to-LAB transformation function. As shown in the figure, the transformed boundary is an irregular shape The next step includes a modified scan-line polygon-filling algorithm that is used in order to fill the boundary with training samples. The training samples are trained by GLA to produce a codebook. The codebook is a set of *n* codebook vectors, which are uniformly distributed in the target color space. Each codebook vector occupies an area (volume in 3D), called a cell. Given a cell, the distance between any point within the cell and the codebook vector of the cell is less than distance between the point and other codebook vectors. The Voronoi diagram shown in the diagram conveys this relationship. And finally, a lookup table is generated to map from original color space to index the cell for faster histogram generation in the next phrase.

Boundary Sample Generation

Uniform samples are not taken in the original color space for initialization because these samples may not be uniformly distributed in the target color space after color transformation. Instead, the boundary of the original color space is used as initialization.

First, 6 flat surfaces of the RGB cube are generated. In each surface, each axis is uniformly discretize to generate a set of 3-dimensional coordinates $S = \{ c_1, c_2, \ldots c_m \mid c_i \in C_{rgb} \}$ as samples. The algorithm is shown as follows:

```
function Boundary SampleGeneration(delta)
begin
S = ∅;
for i := 0.0 to 1.0 step delta
     for j := 0.0 to 1.0 step delta
          S = S ∪ [0, i, j] ∪ [i, 0, j] ∪ [i, j, 0] ∪ [1, i, j] ∪ [i, 1, j] ∪ [i, j, 1];
     return S;
end;
```

where delta controls the number of boundary samples.

Color Space Transformation

Let f: C_{rgb} -> C' is the transformation from RGB to another color space C', the coordinates are transformed to $S' = \{ f(c_i) \mid c_i \in S \}$. Figure 5-3 depicts the boundary of CIELUV color space.

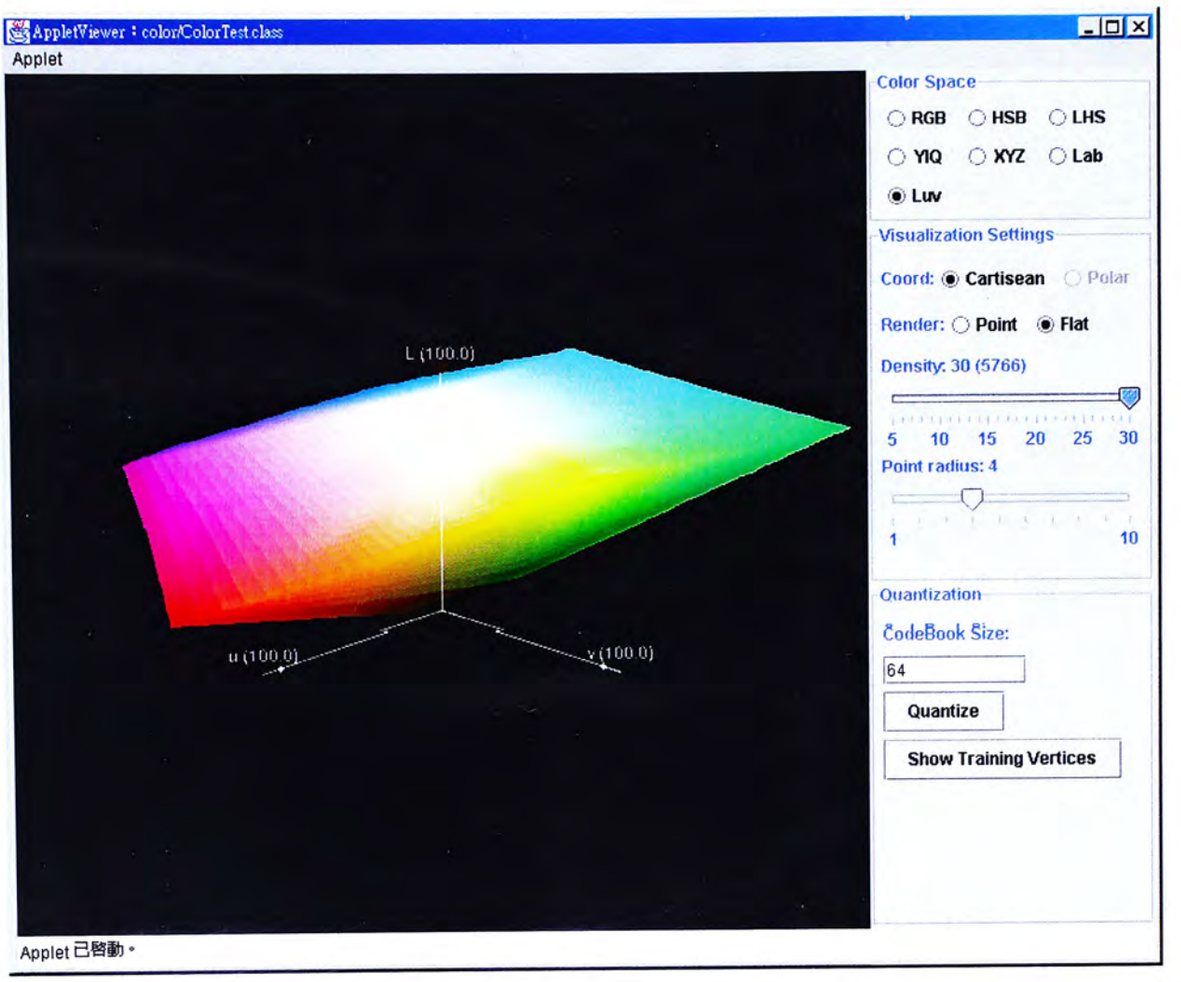


Figure 5-3: CIELUV color space visualized by the 6 surfaces.

Training Sample Generation

To create training samples in the target color space, the scan-line polygon-filling algorithm is extended to 3-dimensional space in order to generate uniform training samples within the target color space. The following shows the algorithm.

```
function ScanLine(S', resolution)
begin
    x_min := min { c_x | c ∈ S' };    x_max := max { c_x | c ∈ S' };    x_length := x_max - x_min;
    y_min := min { c_y | c ∈ S' };    y_max := max { c_y | c ∈ S' };    y_length := y_max - y_min;
    z_min := min { c_z | c ∈ S' };    z_max := max { c_z | c ∈ S' };    z_length := z_max - z_min;
    scale := resolution / max { x_length, y_length, z_length };

    z_min[0 ... (x_length * scale), 0 ... (y_length * scale)] := { ∞, ∞, ... };
```

```
    zmax[0 ... (xlength * scale), 0 ... (ylength * scale)] := { -∞, -∞, ... };

    for each c ∈ S'
    begin
        x = round(cx * scale);    y = round(cy * scale);    z = round(cz * scale);
        zmin[x, y] = min (zmin[x, y], z);
        zmax[x, y] = max (zmax[x, y], z);
    end;

    T = ∅;
    for x := 0 to xlength * scale
    for y := 0 to ylength * scale
        if zmin[x, y] ≠ ∞ and zmax[x, y] ≠ -∞ then
            for z := zmin[x, y] to zmax[x, y]
                T = T ∪ [x, y, z];
    return T;
end;
```

where resolution is the number of training samples in the maximum length axis.

The result of the ScanLine algorithm is shown in figure 5-4.

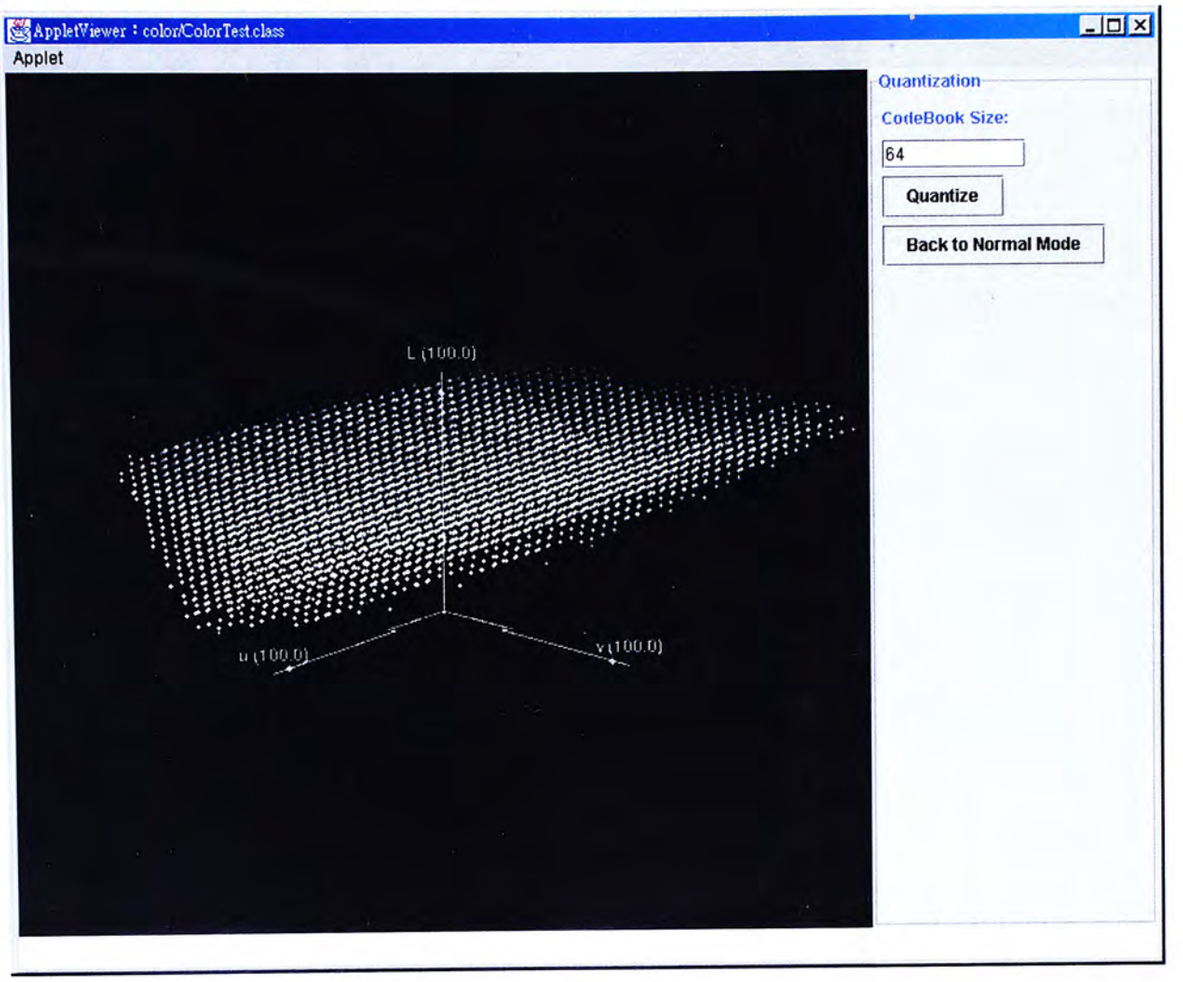


Figure 5-4: Uniform training samples generated by ScanLine

Passing the training set T, dimension n of the histogram, and a threshold Δt to the GLA algorithm, the codebook is generated. Threshold Δt is used to control the accuracy of uniformity. The smaller the Δt, the greater is the accuracy. The GLA algorithm is shown as follows.

```
function GLA(T, n, Δt)
begin
     Initialize an n-dimensional codebook C1 with random values;
     t1 := 0;
     m := 1;
loop:
     Cm+1 := Lylod_Iteration(T, Cm, n);
     tm+1 := AverageDistortion(T, Cm+1, n);

     if |tm+1 - tm| > Δt then
     begin
          m = m + 1;
          goto loop;
     end;

     return cm+1;
end;

function Lylod_Iteration(T, C={y1, y2, …, yn}, n)
begin
     for i := 1 to n
     begin
          // Step1: Partition the training set into cluster sets Ri
          Ri={ x∈T : d(x, yi) ≤ d(x, yj); all j≠i};
          // Step2: Compute the centriod for the cluster sets to obtain new codebook
          centi = cent(Ri) ;
     end;
     return { cent1, cent2, ... , centn };
end;

function AverageDistortion(T, C={y1, y2, …, yn}, n)
begin
     for j := 1 to n
          dj := min_{xi∈T} d(xi, yj);
     return mean(d1,d2,…,dn) ;
end;
```

The final quantization results have been shown in figure 5-5 and 5-6 for two different views.

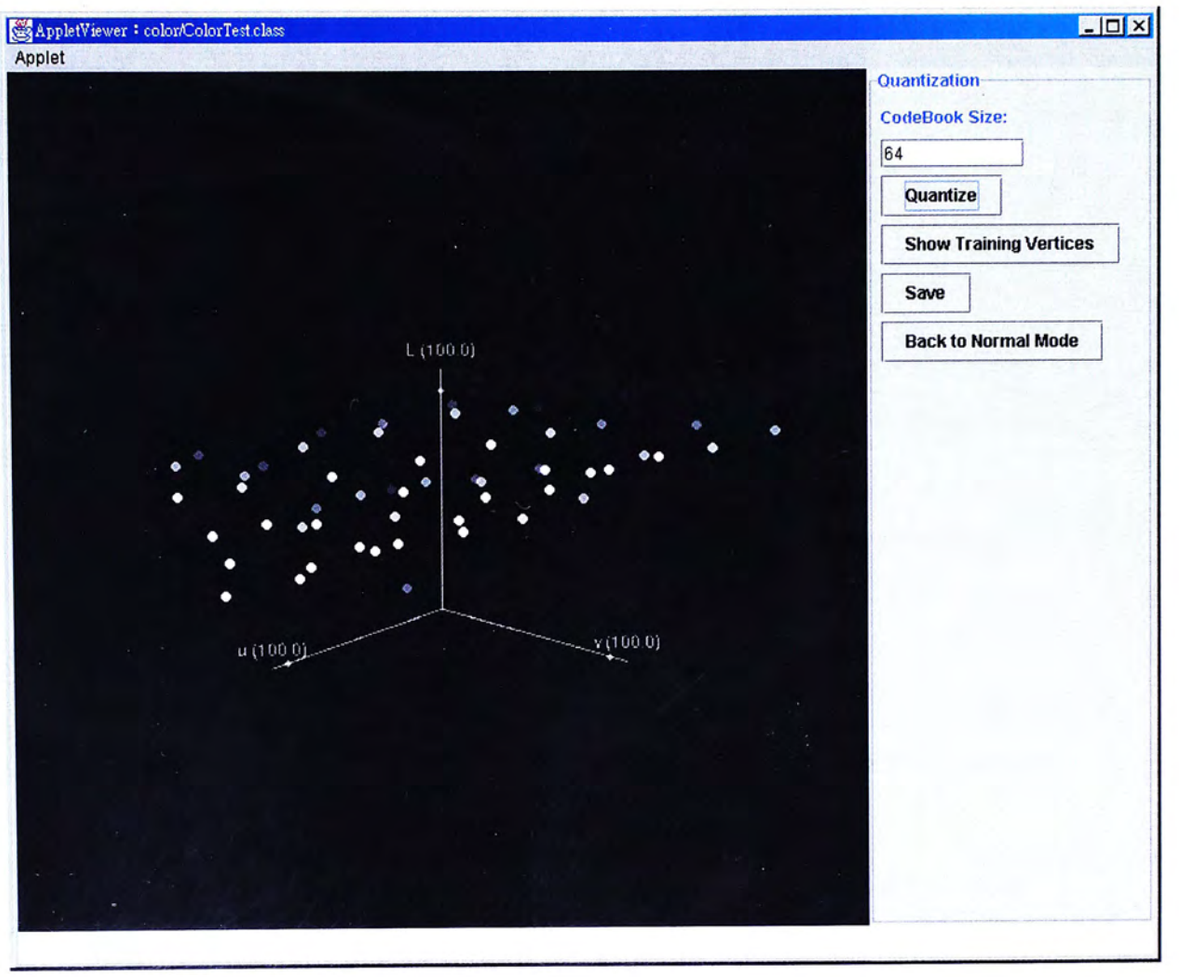


Figure 5-5: 64 cookbook vectors generated by GLA
(Brightness visualizes the distance from the view-point)

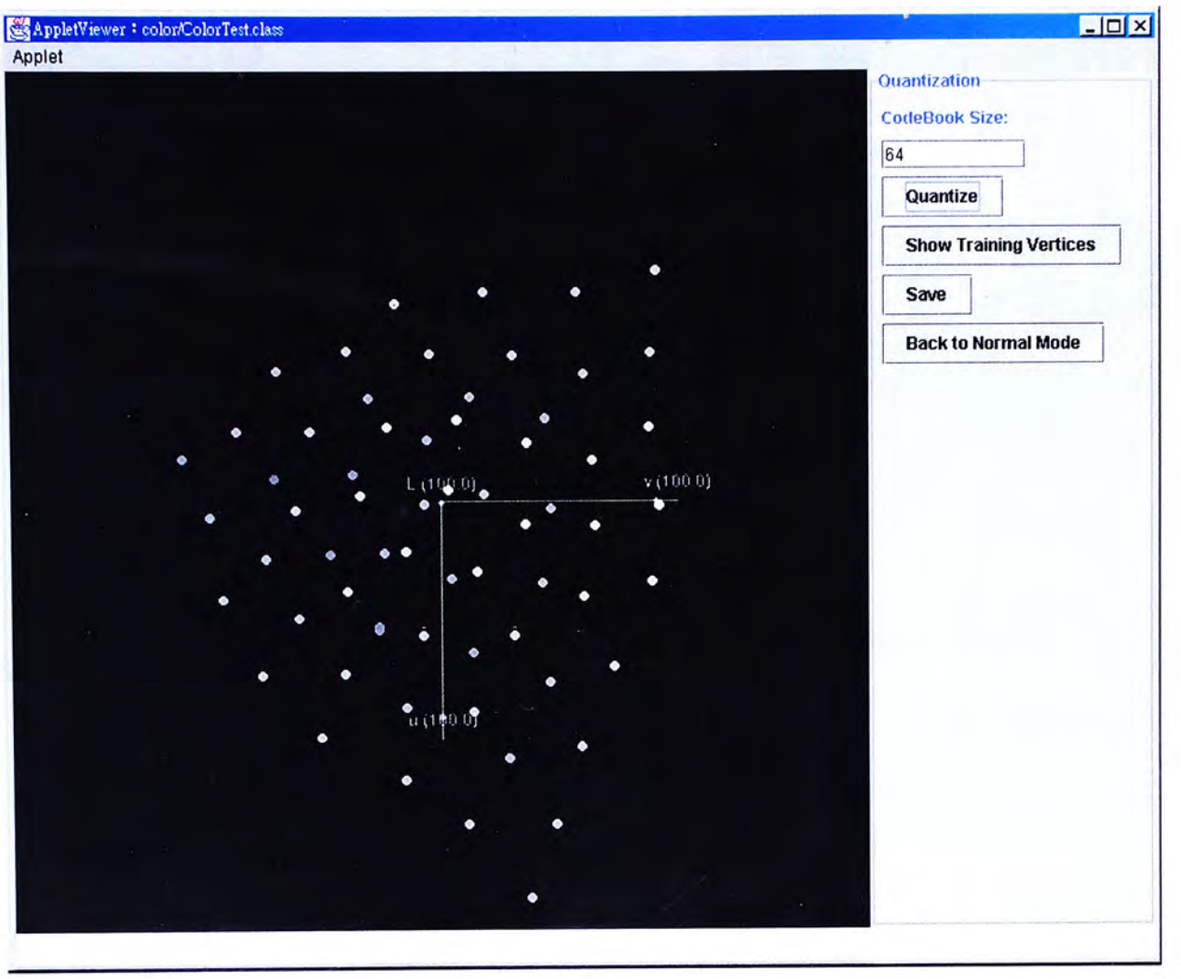


Figure 5-6: The same codebook vectors viewed from the top

(Brightness visualizes the distance from the view-point)

The generated codebook can be used to tessellate the color space in d cells. With regards to the Voronoi diagram, for any coordinate $\boldsymbol{p}$ in the boundary of cell k, $\boldsymbol{p}$ has a minimal Euclidean distance to k among all cells. Therefore, counting the number of colors in an image that falls in the cells generates the color histogram of the image.

Lookup Table Generation

To accelerate the processing of histogram generation, lookup table g is generated for the purpose of mapping each discrete color in RGB to the index of cell.

Let $K=\{k_1, k_2, \ldots, k_n\}$ be the codebook, then

$g(c) = \arg\min \| f(c) - k_i \|$ is the mapping from RGB to index of cell (bin) (5-1)

The physical size of the lookup table g depends on the resolution of the RGB color space and the size of codebook. Typically, each color channel is 8-bit and the number of codebook vectors is less than 256, which generates a 16MB lookup table.

5.1.2. Histogram Generation Phrase

A Color Histogram $H(M)$ of image M is a 1-D discrete function representing the probabilities of occurrence of colors in images, which is typically defined as:

$$H(M) = [h_1, h_2, \ldots, h_n] \quad (5\text{-}2)$$
$$h_k = \frac{n_k}{N} \qquad k = 1,2,\ldots,n$$

where N is number of pixels in image M and n_k is the number of pixels with image value k. The division normalizes the histogram such that:

$$\sum_{k=1}^{n} h_k = 1.0 \quad (5\text{-}3)$$

As a lookup table g has been defined to transform the color space to the histogram space, the equation (5-2) can be modified as:

$$h_i = \frac{1}{N} \sum_{j=1}^{N} \begin{cases} 1 & i = g(c_j) \\ 0 & otherwise \end{cases} \quad (5\text{-}4)$$

Since the lookup table g pre-calculated the color space transformation function f, the

run-time complexity of histogram generation phrase for any color space is the same as the traditional histogram approach.

5.2. Experiment

Two types of experiments have been conducted in order to evaluate quantitative performance of this innovative approach.

5.2.1. Test Database

For this study, the image collection from UC Berkeley's digital library project has been used [Berkeley].

5.2.2. Evaluation Methods

This new algorithm has been compared to the traditional approach by two metrics. Firstly, indexing effectiveness is a measurement of the discriminating power of the feature by empirical results in a test database. Secondly, human evaluation is performed to compare the precision-recall performance between the two algorithms.

Indexing Effectiveness

In image retrieval task, the system compares a query image with all images in the database. Subsequently each image in the database is associated with a similarity value. Histogram capacity is a measure on the similarity distribution defined in [Stricker94]:

Definition 1 Given an n-dimensional histogram space H, a metric d on H and a distance threshold t, the capacity C of H is given by the maximal number of t-different histograms that fit into H.

Two metrics, L1-norm and L2-norm, are used in this experiment. The metrics have been normalized in [0, 100], formulated as:

$$d_{L_1}(\mathbf{x},\mathbf{y}) = 50 \cdot \sum |\mathbf{x}_i - \mathbf{y}_i| \tag{5-5}$$

$$d_{L_2}(\mathbf{x},\mathbf{y}) = 100 \cdot \sqrt{\frac{1}{2}\sum (\mathbf{x}_i - \mathbf{y}_i)^2} \tag{5-6}$$

An *empirical capacity curve C(t)* can be computed by all image couples in a database [Brunelli99]. In [Brunelli00], the *indexing effectiveness* of a histogram space is defined as:

$$\varepsilon = \int tC(t)dt \tag{5-7}$$

The indexing effectiveness ε is the average dissimilarity between all image couples.

User Evaluation

An image retrieval test has been conducted by means of user evaluation. In this experiment, 18 subjects have been selected, all of whom are undergraduates in the Department of System Engineering and Engineering Management of the Chinese University of Hong Kong. The experiment used 2346 images in the collection, 50 of which were randomly selected as the query images. Each subject was required to evaluate whether or not each image in the pre-set query result was relevant to the query image (Figure 5-7). The pre-set query result was generated by the union of the top 50 query results using each color space, plus 10% noise. A total of 176 queries were evaluated by the subjects.

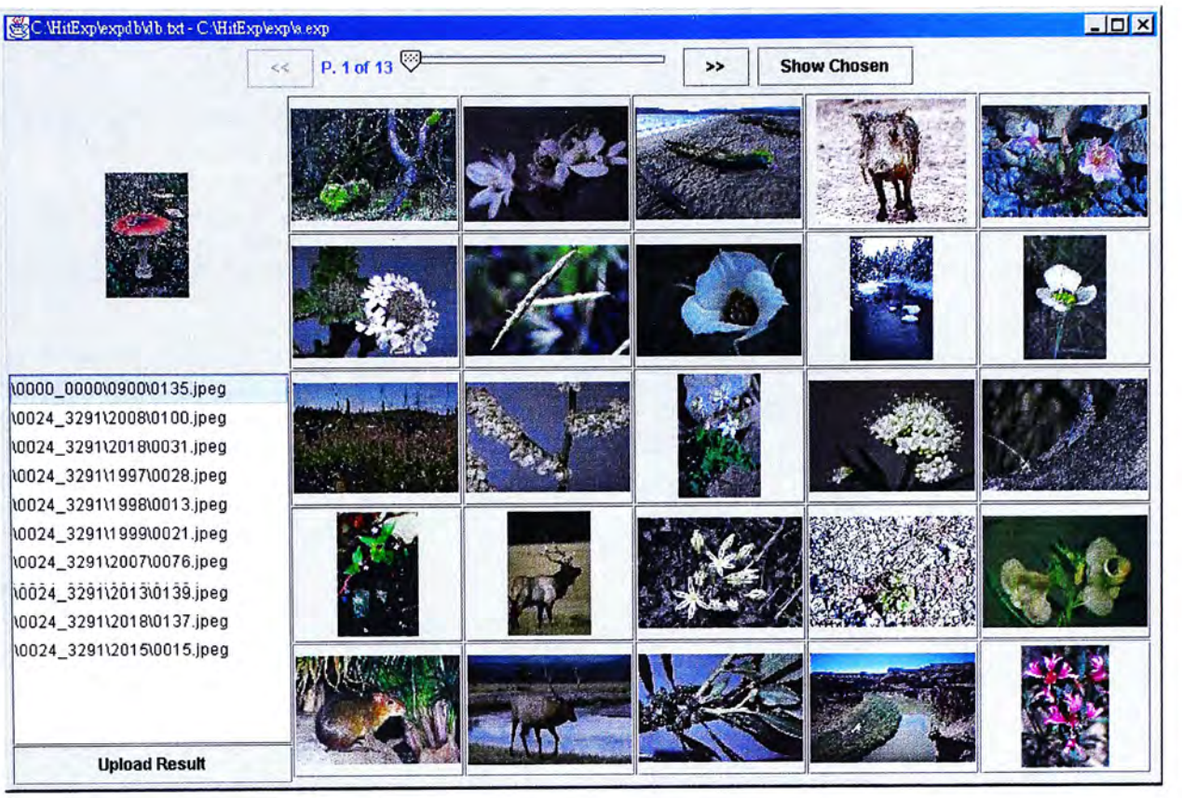


Figure 5-7: User interface of the experiment software.

(Image on the left-top is the query image. Images on the right are the preset results.)

At that stage, the user evaluation was compared against the results queried by each color space. Precision and recall notation was used to measure the performance.

$$\text{recall} = \frac{\text{relevant retrieved image}}{\text{total number of relevant image in database}} \tag{5-8}$$

$$\text{precision} = \frac{\text{relevant retrieved image}}{\text{total number of retrieved image}} \tag{5-9}$$

5.2.3. Results and Discussion

Indexing Effectiveness

In this experiment, the indexing effectiveness was calculated for 9 histogram spaces and the two metrics, using 2346 images in the UC Berkeley image collection. Table

5-1 and 5-2 shows the experiment results. All histogram spaces have 64 bins. The key "HSV_4_4_4_L1" represents a HSV color space, of which each axis is quantized with 4 equal divisions. The keys prefixed with "Q" represent the specific color space that is quantized using the algorithm in section 2. The empirical capacity curves of various histogram spaces are depicted in the following two graphs.

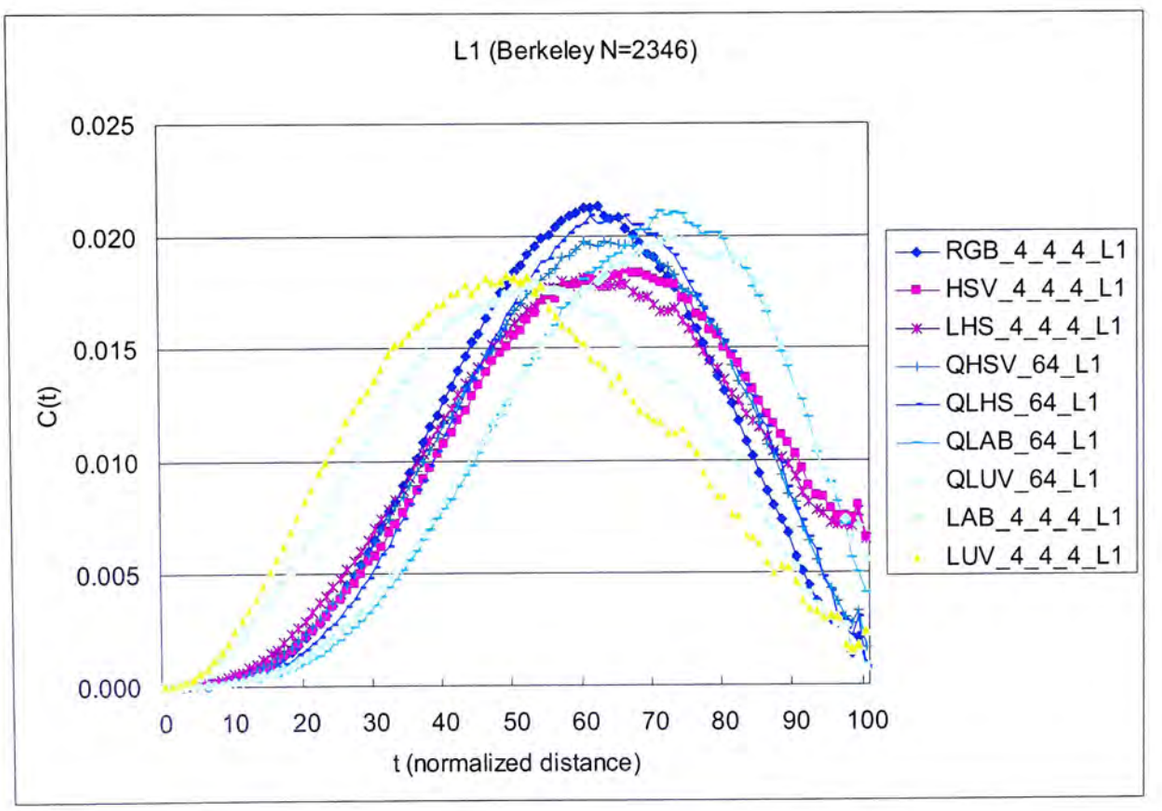


Figure 5-8: The empirical capacity of various histogram spaces for L1 norm

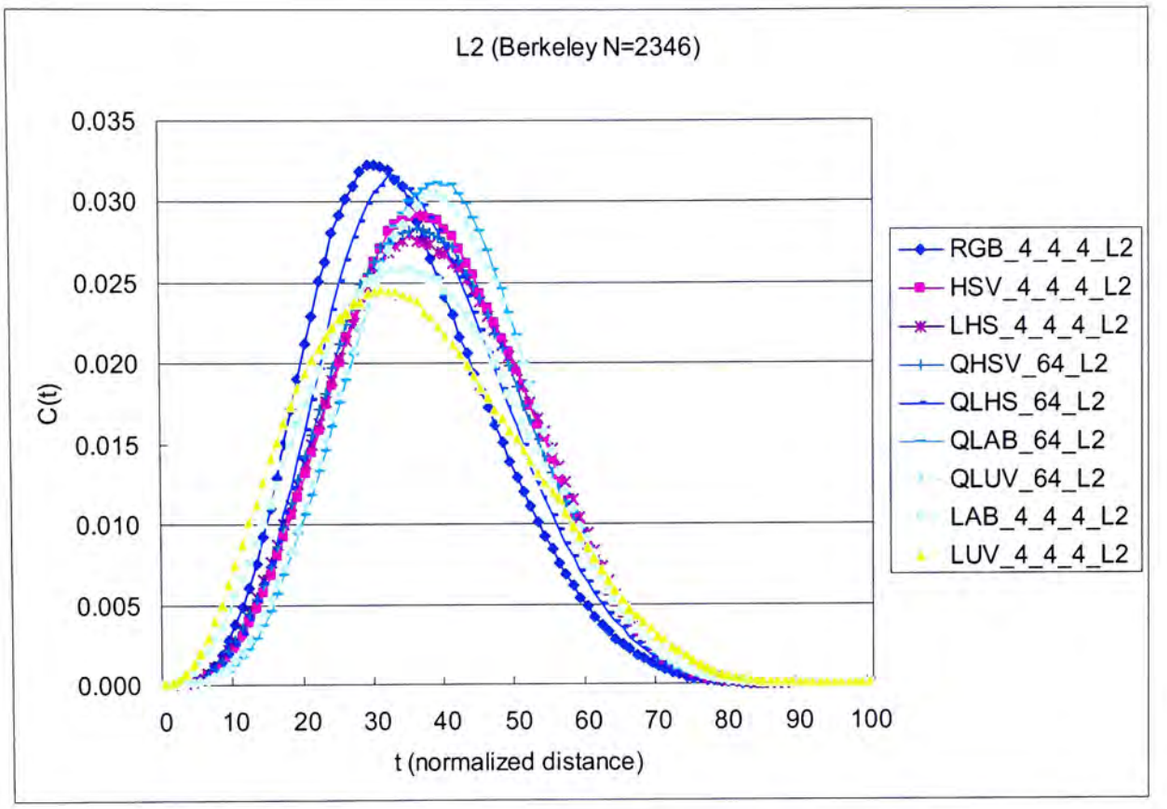


Figure 5-9: The empirical capacity of various histogram spaces for L2 norm

The graphs are the density distribution of the distances between all image couples[2]. The x-axis is the normalized distance and the y-axis is the empirical capacity of the histogram, that is, the frequency of image couples having distance *t*. If the distance is large in a histogram, indexing is effective. In both figures, QLAB_64 and QLUV_64 are located on the right of the other histogram spaces, which indicates their higher capacities in differentiating images.

The indexing effectiveness, as defined in (5-7), is the average dissimilarity between all image couples. The empirical indexing effectiveness is shown in the following two tables.

[2] The number of image couples = N(N-1)/2 = 2750685

Table 5-1: Indexing effectiveness for L1

Key	Effectiveness ε
QLAB_64_L1	66.76
QLUV_64_L1	66.41
HSV_4_4_4_L1	62.84
QLHS_64_L1	61.64
LHS_4_4_4_L1	61.12
QHSV_64_L1	60.82
RGB_4_4_4_L1	59.31
LAB_4_4_4_L1	53.87
LUV_4_4_4_L1	50.80

Table 5-2: Indexing Effectiveness for L2

Key	Effectiveness ε
QLAB_64_L2	39.57
QLUV_64_L2	39.49
LHS_4_4_4_L2	38.40
HSV_4_4_4_L2	38.35
QHSV_64_L2	38.05
LAB_4_4_4_L2	36.46
QLHS_64_L2	36.42
LUV_4_4_4_L2	35.83
RGB_4_4_4_L2	34.11

For indexing effectiveness using L1-norm, color histograms using this specific algorithm (QLAB, QLUV) are higher than some commonly used color histograms, like RGB, HSV, LHS, LAB and LUV. Using L2-norm, the indexing effectiveness of the color histograms using this algorithm (QLAB, QLUV) is also higher than that of the other color histograms. However, for both L1 and L2 norm, the differences between HSV and QHSV, LHS and QLHS are as significant as the differences between QLAB and LAB, QLUV and LUV.

Human Evaluation

Comparison of the results of color histogram was made between this algorithm and those of the traditional ones. The precision-recall relationships are depicted in Figure 5-10 and Figure 5-11.

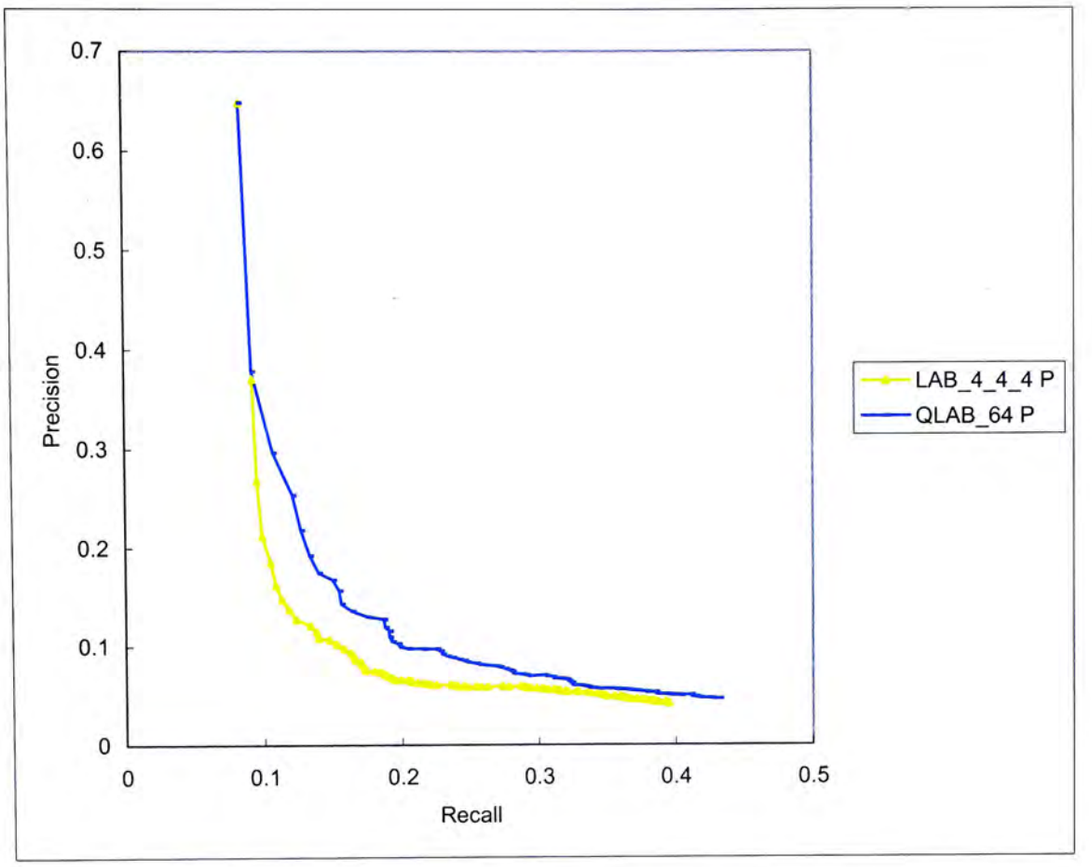


Figure 5-10: Precision-Recall Results for LAB and QLAB

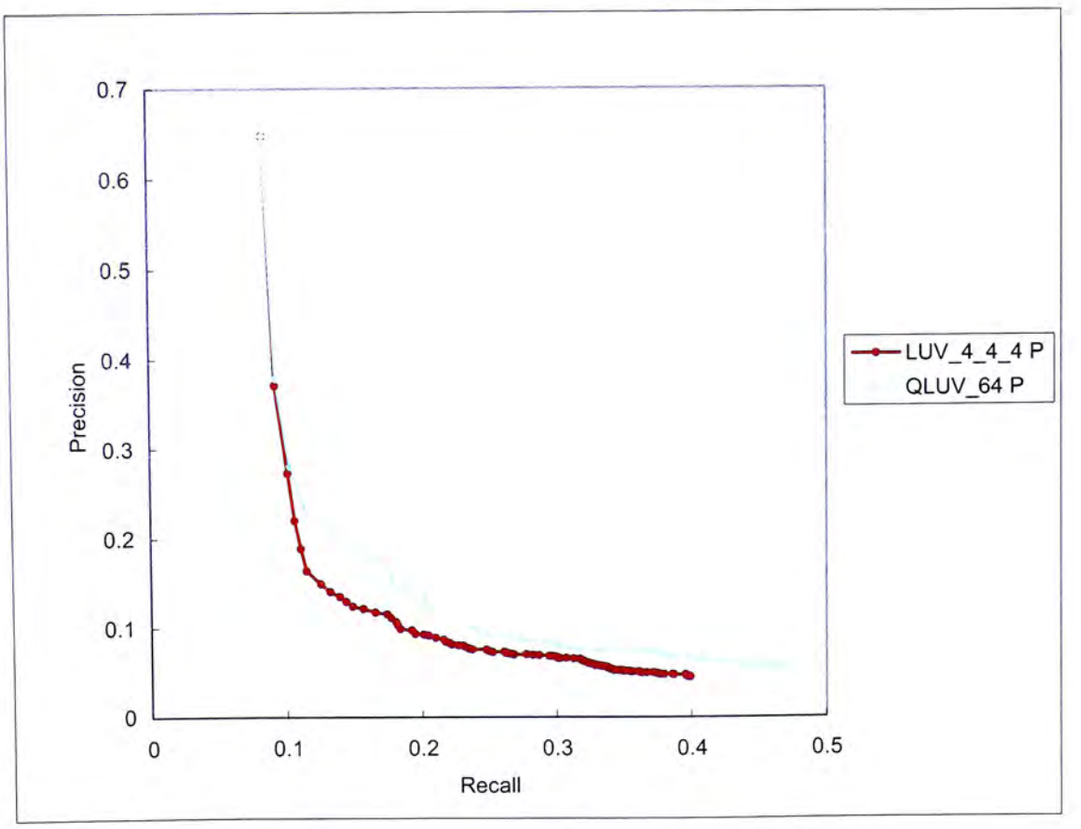


Figure 5-11: Precision-Recall Results for LUV and QLUV

Both graphs show that the histograms using the present algorithm have higher precision over all recall range. That would indicate that this algorithm has a higher performance in human evaluation.

5.2.4. Summary

Empirical results of both indexing effectiveness and human evaluation show that QLAB and QLUV histograms outperform to the traditional approach. The current study has shown that there are very low computational overheads in using this particular algorithm by using a fixed size, pre-generated lookup-table.

6. Relevance Feedback

Relevance feedback is a powerful technique used in traditional text information retrieval systems. According to Riu et. al [Rui98b],

> *"relevance feedback is the process of automatically adjusting an existing query using the information fed-back by the user about the relevance of previously retrieved objects such that the adjusted query is a better approximation to the user's information need."*

6.1. Relevance Feedback in Text Information Retrieval

In text information retrieval models, if the sets of relevant documents (D_R) and non-relevant documents (D_N) are known, the optimal query (Q_{opt}) can be proven to be [Rui97b]

$$Q_{opt} = \frac{1}{N_R}\sum_{i \in D_R} D_i - \frac{1}{N_T - N_R}\sum_{i \in D_N} D_i \tag{6-1}$$

where N_R is the number of documents in D_R and N_T is the number of total documents.

Practically, D_R and D_N are not known in advance. However, approximation to D_R and D_N from user, as D_R' and D_N' can be found. The original query Q can be modified by putting more weights on the relevant terms and fewer weights on the non-relevant terms, as formulated as

$$Q' = \alpha Q + \beta\left(\frac{1}{N_{R'}}\sum_{i \in D'_R} D_i\right) - \gamma\left(\frac{1}{N_{N'}}\sum_{i \in D_N} D_i\right) \tag{6-2}$$

where α, β, and γ are suitable constants, $N_{R'}$ and $N_{N'}$ are the numbers of documents in D'_R and D'_N respectively. As the relevance feedback iteration moves on, Q' will approach Q_{opt}. Experiments show that retrieval performance can be improved considerably by using relevance feedback.

6.2. Relevance Feedback in Multimedia Information Retrieval

Recent research has applied relevance feedback techniques to CBIR systems. For instance, Riu et. al's MARS (Multimedia Analysis and Retrieval Systems) [Rui97a, Rui97b, Rui98a , Rui98b] used an interactive retrieval approach. During the retrieval process, the user's high-level query and perception subjectivity are captured by dynamically updated weights based on user's feedback. This approach bridges the gap between high-level concepts and low-level features using the information of user interactions, to prevent the drawbacks of forcing users to decompose query into feature representation and precisely specify all weights in computer centric approach.

Another CBIR system, ImageRover [Sclaroff97, Taycher97], uses another approach for embedding relevance feedback in their system. ImageRover employs a relevance feedback algorithm that select appropriate Minkowski distance metrics on the fly according to the user's submission of relevant images.

6.3. Relevance Feedback in Visual Thesaurus

The previous reviews show that most research employs relevance feedback in query systems. This study suggest that relevance feedback can also be added to the visual thesaurus approach, which is based on browsing techniques. The basic idea is described as follows.

When a user find some relevant images using the browsing mechanism, he/she can put those images into a relevant bag. System can give feedback to user by visualization of the relevance in the SOM labels. SOM intrinsically cluster relevant images as neighborhoods therefore user can browse nearby labels for relevant images. Relevance feedback enhances this property by visualizing the relevance of a set of relevant images.

To test the idea of embedding relevance feedback in the system, formulations and user-interface prototypes have been constructed. Since in this prototype the system only concerns about relevant images (ignoring irrelevant images) and the system does not emphasize the importance of original query, α=0, β=1, and γ=0 are used in formula 6-2:

$$Q' = \frac{1}{N_{R'}} \sum_{i \in I_{R'}} I_i \qquad (6\text{-}3)$$

That is, the mean of the relevant images.

Then, for each image I in the database, similarity $D(I, Q')$ are evaluated between image i and the query Q' by some metric, for example, L2 norm (2-2).

The major problem is how to visualize the relevance of the labels of SOM. As any image that is similar to Q' but not similar to the other images in the same node should not be omitted, the relevance r_j of each node j is calculated by maximizing the similarity of images in the node:

$$r_j = \max_{i \in R_j} D(Q', I_i) \tag{6-4}$$

where R_j is the set of images in the node.

Then, the approach of using luminance of images to represent this relevance is tested. It is implemented by adjustment of the luminance channel of each label image. The nodes that are most relevant remain unchanged, where nodes that are non-relevant will become dark in the level of non-relevance. However, this approach cannot be applied to images of different luminance range. For example, an image originally dark will have little changes when applying this luminance alternation.

To raise user's attention to nodes with higher relevance and reduce visual complexity, a threshold value λ is employed to trim all nodes with relevance lower than λ. In the prototype, a slide-bar in scale from 0 to 100 is attached in the user interface to let user changes this value interactively.

In figure 6-1, the rightmost column is the newly added relevance bag. Users can drag images to the relevance bag to indicating that the image is relevant to his/her query. After modification to the relevance bag, the system will visualize the relevance in the labels of SOM. In the figure, the threshold λ is 50. Users can pay attention to those labels with relevance higher than threshold. Labels that are under the threshold will be visualized by different level of gray color. In practice, user can set the threshold to a high value first. When the user cannot find the target image or relevant images, he/she can iteratively decrease the threshold.

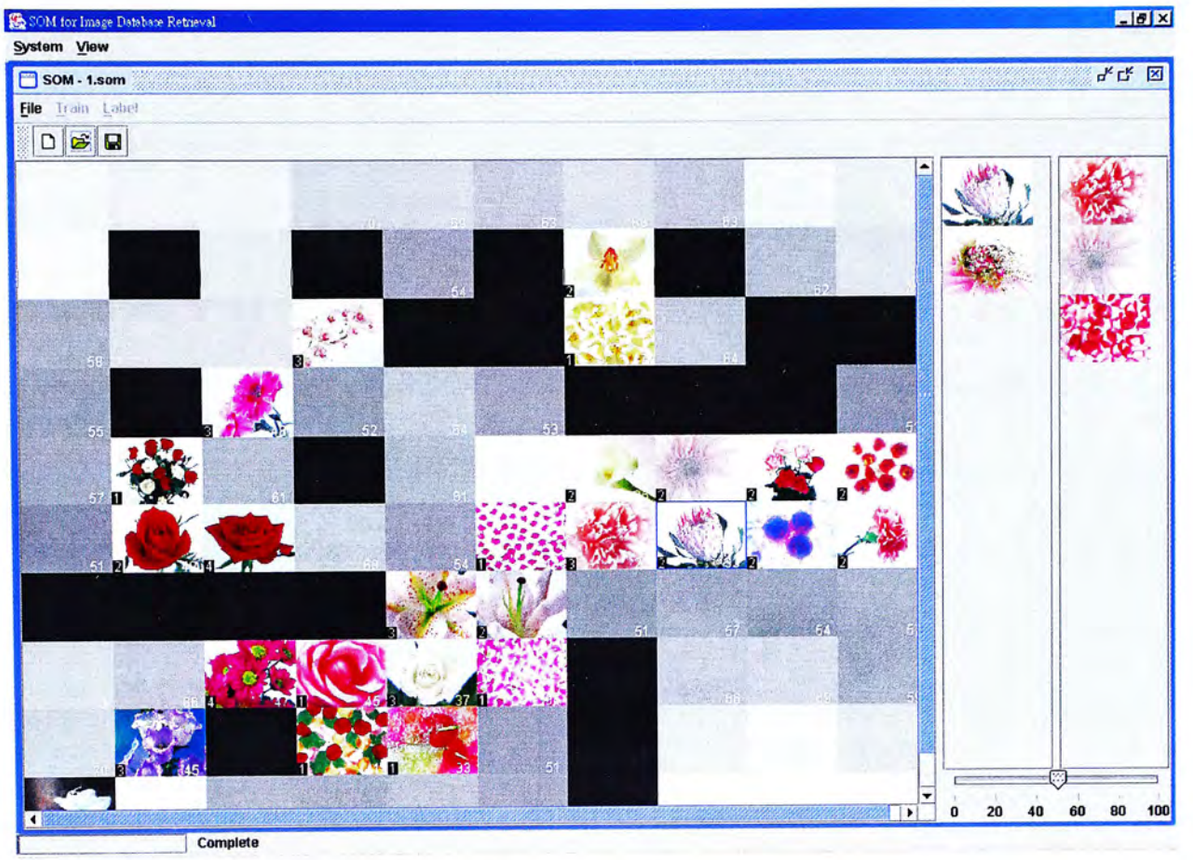


Figure 6-1: Prototype of embedding relevance feedback

Human evaluation studies can be done to evaluate the effectiveness of using relevance feedback as an additional tool to visual thesaurus. Also, it is possible to use non-relevance information. One approach is to add all images that have not been selected as relevance to irrelevance bag. Besides this type of formulation, information of relevance bags can be persisted. This information is subjective evaluation of similar images, which can be used as features or weights during query processing or SOM generation, which similar to MARS.

7. Conclusions

In this thesis, a novel browsing approach for efficient color image retrieval have been proposed. Image feature and SOM are used to visualize a summary of all images in the database. The algorithms and procedures are given in chapter 3. Comparisons of results of SOM trained by different features and labeling methods have been evaluated. Experiments show that color feature and texture feature can be combined to balance the advantages of the two features. Also, SOM labeled by "result- similarity" is most suitable for representing both the map and the images mapped to the nodes.

The human evaluation presented in chapter 4 shows empirical findings to support that The SOM approach outperforms traditional QBE approach by (1) higher efficiency; (2) higher successful rate; and (3) encouraging more queries. The survey results indicate that users are more satisfied in using SOM than QBE in all factors, including general, format, ease of use, and timeliness.

Besides, a general algorithm for quantizing color histogram have been proposed in chapter 5. The algorithm can tessellate non-cubical color spaces, for example, CIELAB (CIE L*a*b*) and CIELUV (CIE L*u*v*), which are more appropriate in CBIR systems because of their perceptually uniform property. Empirical results of traditional approach to this innovative approach have been compared in terms of indexing effectiveness and human evaluation. Findings show that this quantizing algorithm outperforms the traditional algorithm in both evaluation methods.

Finally, an approach to employ relevance feedback in visual thesaurus system have been proposed. A prototype of the user interface is developed to test its applicability.

Human evaluation studies can be done in future to evaluate the effectiveness of using relevance feedback as an additional tool to visual thesaurus.

7.1. Applications

The visual thesaurus approach can be applied to general color image databases as presented in the experiments. However, it can also be applied to other domain-specific image databases, for example, aerial, facial, fingerprints, products and so on, through training on the features extracted from those domains.

Currently, visual thesaurus have been proposed for online product catalog [Yang02a, Yang02c] in which the products can be dynamically and automatically categorized according to low-level image features or product specific features. Major advantages over traditional QBE technique include 1) an exploration of all existing products in a single screen automatically, 2) a support of feature-based product search, 3) ease of use and user-friendliness.

7.2. Future Directions

7.2.1. SOM Generation

SOM is an ideal technique for generating maps for image browsing since it can reflect the relationship among all images in the database. However, there are some restrictions of the current approach using SOM. First, if the number of images is large, the number of images mapped in one node will be large. It is inefficient for user to browse. An immediate derived approach to solve this problem is to generate SOMs for the images mapped to each node. This approach can been applied recursively until the number of images in the node is less than a specific amount. Actually, this approach is similar to the Hierarchical Self-Organizing Map (HSOM) from Zhang et

al. and the Multilayered Self-Organizing Feature Map (M-SOM) from Chen et al. [Han95]. The purposes of these two researches are visual indexing and internet categorization respectively. Also, this kind of SOM variations can be applied to the present research. However, appropriate algorithms for labeling are required.

Another restriction is lack of dynamic maintenance. Inserting or deleting images needs to re-train the SOM. For a normal single layer SOM, additional training iterations can be applied to the existing SOM to adapt the new images. Nevertheless it cannot apply for HSOM/M-SOM approach; modifications of top-level SOM make it necessary to retrain the lower-level SOM. The present study will be continued on this field in order to solve this problem.

7.2.2. Hybrid Architecture

Browsing and searching are not exclusively competitive techniques. Each of them has unique pros and cons over another. Browsing can provide a broad overview of the whole database and then the user can refine the scope of browsing. Searching can generate ranked accurate results by customizable complex queries. Hybrid architecture can combine advantages of both techniques. A simple approach to combine the techniques is to use a common relevant example bag, which stores examples from browsing or searching systems. In browsing system, those relevant examples can be visualized by relevance. In search system, the examples can be used as whole or part of queries. By the interaction between the two systems, user can find images for different purposes and situations. An overview of the system is depicted in Figure 7-1.

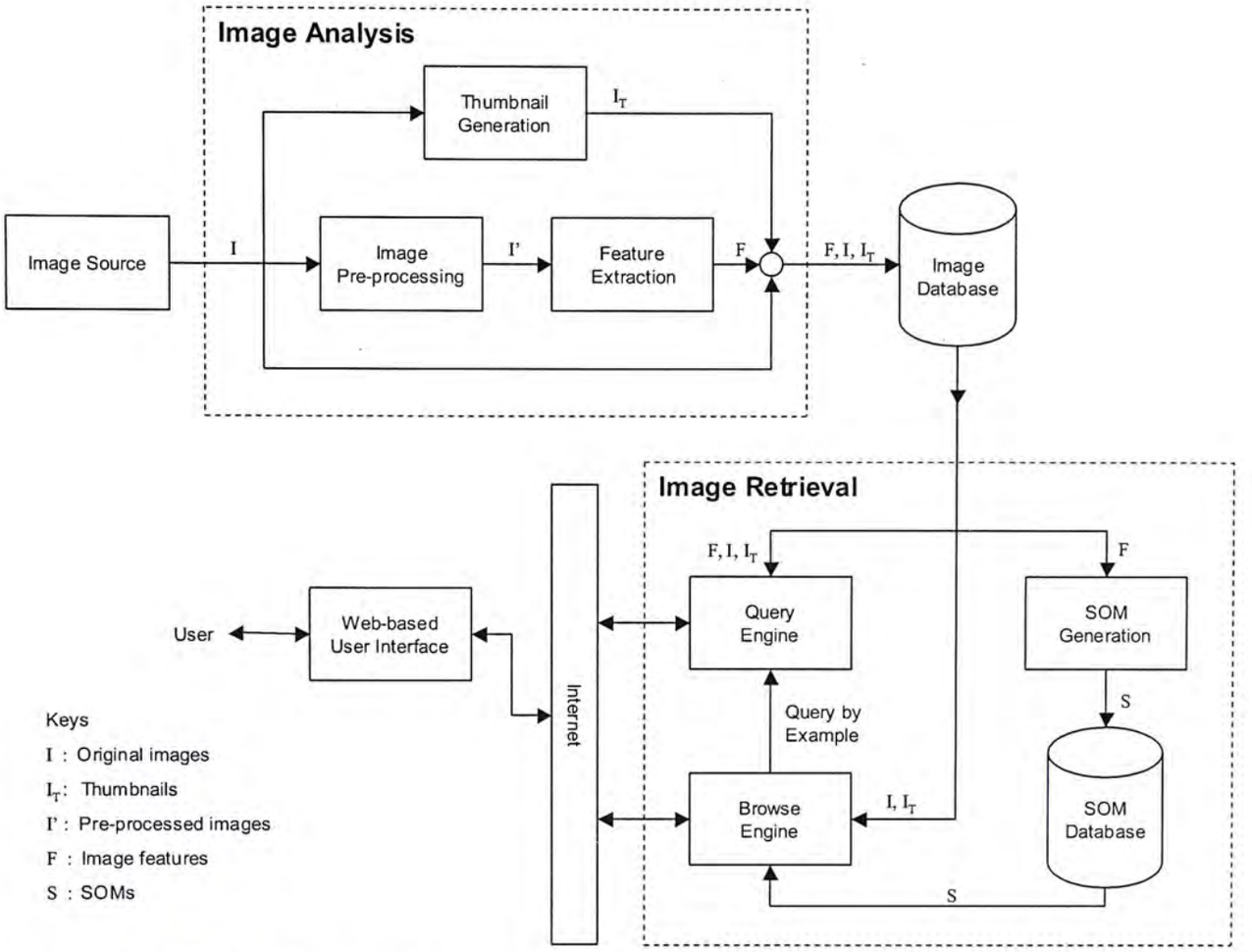


Figure 7-1: Hybrid Architecture

References

[AltaVista] AltaVista Photo Finder, http://image.altavista.com/

[Berkeley] Digital Library Project, University of California, Berkeley, http://elib.cs.ucsb.edu/

[Brunelli00] R. Brunelli and O. Mich, "Image Retrieval by Examples. In IEEE Transactions on Multimedia", Vol. 2, No.3, September 2000.

[Brunelli99] R. Brunelli and O. Mich, "On the Use of Histograms for Image Retrieval", in Proc. ICMCS'99, Florence, June 1999.

[Chen96] H. Chen, C. Schuffels and R. Orwig, "Internet Categorization and Search: A Self-Organizing Approach", *Journal of Visual Communication and Image Representation*, Vol. 7, No.1, pp. 88-102, March 1996.

[Chen99a] J.-Y. Chen, C.A. Bouman and J. C. Dalton, "Active Browsing using Similarity Pyramids", *Proc. Of IS&T / SPIE Conference on Storage and Retrieval for Image and Video Database VII*, vol. 3656, pp. 144-154, (San Jose, California), January 1999.

[Chen99b] C. C. Chen and C. C. Chen, "Filtering Methods for Texture Discrimination", *Pattern Recognition Letters*, vol. 20, pp. 793-790, 1999.

[Craver98] S. Craver, B.-L. Yeo, and M.M. Yeung, "Image browsing using data structure based on multiple space-filling curves". To appear in the *Procceedings of the Thirty-Sixth Asilomar Conference on Signals, Systems, and Computers*, pp. 155-166 (Pacific Grove, CA), November 1-4 1998.

[Doll88] W. J. Doll, and G. Torkzadeh, "The Measurement of End-user Computing Satisfaction", *MIS Quarterly*, pp259-274, June 1988.

[Flickner95] M. Flickner, H. Sawhney, and W. Niblack et al., *Query by image and video content: The QBIC system*, IEEE Computer, September 1, 1995, 23--32.

[Frankel96] C. Frankel, M. Swain, and V. Athitsos, *Webseer: An Image Search Engine for the World Wide Web*, Technical Report TR-96-14, CS Department, Univ. of Chicago, 1996.

[Gersho92] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression,* Kluwer Academic Publishers, Boston, 1992.

[Google] Google, http://www.google.com/

[Gupta97] A. Gupta, R. Jain, "Visual information retrieval," *Comm. Assoc. Comp. Mach.*, vol. 40, no. 5, pp. 70-79, May 1997.

[Hafner95] J. Hafner, H.S. Sawhney, W. Equitz, M. Flicker, and W. Niblack, "Efficient Color Histogram Indexing for Quadratic Form Distance Functions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 7, July 1995, pp. 729-736.

[Han95] K. A. Han, J. C. Lee and C. J. Hwang, "Image Clustering using Self-organizing feature map with Refinement", *Proceedings IEEE International Conference on Neural Networks*, Vol. 1, pp. 465-469, 1995.

[Haradar97] S. Haradar, Y. Itoh, and H. Nakatani, Interactive image retrieval by natural language, *Optical Engineering* 36(12), pp3281-3287, 1997.

[Kohonen84] T. Kohonen, "Self-Organization and Associative Memory", *Springer Series in Information Science*, vol. 8, 1984.

[Kohonen95] T. Kohonen, "Self-Organizing Maps", *Springer Series in Information Science*, vol. 30, 1995.

[LaCascia98] M. La Cascia, S. Sethi, and S. Sclaroff., "Combining textual and visual cues for content-based image retrieval on the world wide web", *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 24--28, Santa Barbara, California, 1998. IEEE Computer Society.

[Manjunath96]B. S. Manjunath and W. Y. Ma, "Texture Features for Browsing and Retrieval of Image Data", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 18, pp. 837-842, August 1996.

[Manjunath97]W. Y. Ma and B. S. Manjunath., "NETRA: A toolbox for navigating large image databases", *IEEE International Conference on Image Processing*, 1997.

[Niblack93] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin, "The QBIC Project: Querying Images By Content Using Colors, Texture and Shape," Storage and retrieval for image and video database : 2-3 February 1993, San Jose, California, *Proc of SPIE – the International Society for Optical Engineering: Vol. 1908,* c1993, pp173-187.

[Niblack98] W. Niblack, Z. Zhu, J.L. Hafner, T. Breuel, D. Ponceleon, D. Petkovic, M. Flickner, E. Upfal, S.I. Nim, S. Sull, B. Dom, B.L. Yeo, S. Srinivasan, D. Zivkovic, and M. Penner, "Updates to the QBIC System," Storage and retrieval for image and video database VI : 28-30 January 1998, San Jose, California, *Proc. of SPIE – the international Society for Image Science and Technology,* c1997, pp. 150-161.

[Pentland94] A. Pentland, R. Picard, and S. Sclaroff, "Photobook: Content-based Manipulation of Image Databases", *SPIE Storage and Retrieval for Image and Video Databases II*, number 2185, San Jose, CA., February 1994.

[Rui97a] Y. Rui, T. S. Huang, S. Mehrotra, and M. Ortega, "A relevance feedback architecture in content-based multimedia information retrieval systems", *In Proc of IEEE Workshop on Content-based Access of Image and Video Libraries*, 1997.

[Rui97b] Y. Rui, T. S. Huang, and S. Mehrotra, "Content-based image retrieval with relevance feedback in MARS", *In Proc. IEEE Int. Conf. on Image Proc.*, 1997.

[Rui98a] Y. Rui, T. Huang, and S. Mehrotra. *Relevance Feedback Techniques in Interactive Content-Based Image Retrieval*. In Storage and Retrieval for Image and Video Databases (SPIE), pages 25--36, San Jose, California, USA, Jan. 1998.

[Rui98b] Y. Rui, T. S. Huang, M. Ortega, S. Mehrotra, "Relevance feedback: A power tool in interactive content-based image retrieval", *IEEE Trans. on Circuits and Systems for Video Technology* 8(5), pp644-655, Sep. 1998.

[Rui99] Y. Rui, T.S. Huang and S-F. Chang, "Image Retrieval: Current Techniques, Promising Directions, and Open Issues", *Journal of Visual Communication and Image Representation*, v 10, n 1, p 39-62, March 1999.

[Sclaroff97] S. Sclaroff, L. Taycher, and M. La Cascia, "Imagerover: A content-based image browser for the world wide web", in *Proc. of IEEE Workshop on Content-based Access of Image and Video Libraries*, June 1997.

[Smith96] J. R. Smith, S.-F Chang, "Tools and techniques for color image retrieval", in *Storage & Retrieval for Image and Video Databases IV* , vol. 2670 of IS&T/SPIE Proceedings, San Jose, CA, USA , Mar. 1996, pp426-437.

[Smith97] J. R. Smith, "Integrated Spatial and Feature Image Systems: Retrieval, Analysis and Compression", PhD thesis, Graduate School of Arts and Science, Columbia University, 1997.

[Stricker94] M. Stricker and M. Swain, "The Capacity of Color Histogram Indexing", in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp704-708, 1994.

[Swain91] M. Swain and D. Ballard, "Color Indexing", *International Journal of Computer Vision*, 7:1, pp 11-32, 1991.

[Swain99] M. Swain, "Searching for Multimedia on the World Wide Web", Techincal Report CRL 99/1, Camridge Research Laboratory, http://crl.research.compaq.com/publications/techreports/abstracts/99_1.html, 1999.

[Taycher97] L. Taycher, M. L. Cascia, and S. Sclaro, "Image digestion and relevance feedback in the Imagerover WWW search engine", in *Proceedings of International Conference on Visual Information*, San Diego, CA, December 15-17 1997.

[Yang99] C. C. Yang, and M. C. Chan, "Color Image Retrieval Based on Textural and Chromatic Features," Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Tokyo, Japan, October 12-15, 1999.

[Yang01] C. C, Yang, and M. K. Yip, "Visual Thesaurus for Color Image Retrieval using Self-Organizing Map," Proceedings of the 7th International Conference on Information Systems Analysis and Synthesis, Orlando, July 22-25, 2001.

[Yang02a] C. C. Yang, S. H. Kwok, and M. Yip, "Image Browsing for Infomediaries," Proceedings of the Thirty-fifth Hawaii International Conference on System Sciences, Hawaii, January 7-10, 2002.

[Yang02b] C. C. Yang, M. K. Yip, "Quantization of Color Histograms using GLA," *Proceedings of the SPIE International Conference on Electronic Imaging and Multimedia Technology*, Photonics Asia, Shanghai, China, October 14-18, 2002.

[Yang02c] C. C. Yang, S. H. Kwok, and M. K. Yip, "Image Browsing of Feature-based Products," *Proceedings of the SPIE International Conference on Electronic Imaging and Multimedia Technology*, Photonics Asia, Shanghai, China, October 14-18, 2002.

[Zhang95] H. Zhang and D. Zhong, "A Scheme for Visual Feature based Image Indexing", in *Proc. Of IS&T / SPIE Conference on Storage and Retrieval for Image and Video Database III*, vol. 2420, pp. 36-46, (San Jose, CA), February 9-10 1995.